

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
ESCUELA POLITÉCNICA SUPERIOR DE ELCHE
MASTER UNIVERSITARIO EN ROBÓTICA



CREACIÓN DE MAPAS VISUALES TOPOLÓGICOS MEDIANTE AGRUPAMIENTO DE ESCENAS

TRABAJO FIN DE MASTER
CURSO ACADÉMICO 2016-2017

AUTOR: LAURA RODRÍGUEZ GOMIS
TUTOR: LUIS PAYÁ CASTELLÓ

Índice

Capítulo 1. Introducción	1
1.1. Robótica móvil.....	1
1.2. Objetivo	2
1.3. Contenidos	3
 Capítulo 2. Descriptores de Apariencia Visual Global.....	5
2.1. Propiedades de los descriptores	6
2.2. Descriptores basados en la Transformada Discreta de Fourier.	6
2.2.1. Firma de Fourier	7
2.3. Descriptores basados en Histogramas de Orientación del Gradiente	8
2.4. Descriptores basados en Gist	10
 Capítulo 3. Creación de mapas	13
3.1. Tipos de mapas.....	13
3.2. Etapas en la creación de mapas y localización.....	17
3.3. Creación de mapas topológicos	17
3.3.1. Creación de mapas empleando la Firma de Fourier	17
3.3.2. Creación de mapas empleando HOG	17
3.3.3. Creación de mapas empleando Gist.....	18
3.4. Métodos para la creación de mapas topológicos	18
 Capítulo 4. Agrupamiento o clustering de imágenes.....	19
4.1. Clustering jerárquico	20
4.1.1. Función pdist	20

4.1.2.	Función linkage.....	21
4.1.3.	Función dendrogram	22
4.1.4.	Función Cluster.....	24
4.1.5.	Verificación de precisión	25
4.1.6.	Verificación de disimilitud	28
4.2.	Clustering espectral.....	29
4.2.1.	Función gen_nn_distance	29
4.2.2.	Función sc.....	30
Capítulo 5. Análisis de los resultados obtenidos		31
5.1.	Base de datos empleada	31
5.2.	Equipo empleado	33
5.3.	Adaptación de las imágenes.....	33
5.4.	Estudio del coste computacional	35
5.5.	Resultados obtenidos para una selección de escenas	37
5.6.	Resultados obtenidos para el conjunto completo de escenas.....	49
5.6.1.	Primera fase: Clustering jerárquico.....	49
5.6.2.	Segunda fase: Clustering espectral	69
5.6.3.	Única fase: Clustering espectral	73
Capítulo 6. Conclusiones		85
Capítulo 7. Referencias.....		87

Índice de Figuras

Figura 1.1. Diferentes aplicaciones de robots móviles	2
Figura 2.1. Transformada de Fourier 2D de una escena. (a) Imagen original, (b) Módulo de la Transformada de Fourier, (c) Fase de la Transformada de Fourier	7
Figura 2.2. (a) Imagen panorámica inicial, (b) componente horizontal, (c) componente vertical, (d) módulo del gradiente y (e) fase del gradiente.....	9
Figura 2.3. Histograma de orientación.....	9
Figura 2.4. Pirámide de imágenes Gaussiana.....	10
Figura 2.5. Filtros de Gabor.....	11
Figura 2.6. Filtrado de la imagen mediante filtros de Gabor	11
Figura 2.7. Agrupación de píxeles en bloques para reducir la dimensión de la imagen.....	11
Figura 3.1. Representación de ocupación	13
Figura 3.2. Representación geométrica	14
Figura 3.3. Representación mediante landmarks	14
Figura 3.4. Grafo de vistas.....	15
Figura 3.5. Grafo tipo ruta.....	15
Figura 3.6. Representación por capas	16
Figura 3.7. Organización jerárquica	16
Figura 3.8. Patchwork.....	16
Figura 4.1. Dendrograma obtenido para un conjunto de 10 imágenes.....	23
Figura 4.2. Agrupación mediante dendrograma	24
Figura 5.1. Plano en planta de la ruta recorrida. Language Technology Laboratory, Centro de Investigación Alemán de Inteligencia Artificial en Saarbrücken, Alemania	32
Figura 5.2. Robot móvil PeopleBot.....	33
Figura 5.3. Sistema de visión omnidireccional	33

Figura 5.4. Imagen en formato omnidireccional.....	34
Figura 5.5. Imagen en formato panorámico RGB.....	34
Figura 5.6. Imagen en formato panorámico blanco y negro.....	34
Figura 5.7. Conjunto reducido de imágenes	38
Figura 5.8. Mapa en planta obtenido empleando Gist con acierto del 81.95%.....	68
Figura 5.9. Límites de cada estancia	68
Figura 5.11. Límites de cada estancia	69
Figura 5.10. Resultado obtenido para 2 clusters	69
Figura 5.12. Resultado obtenido para 4 clusters	70
Figura 5.13. Mapa en planta con acierto del 88.72%.....	71
Figura 5.14. Mapa en planta ideal de acierto 100%.....	71
Figura 5.15. Imágenes de las 3 zonas de la cocina. Resultado del clustering espectral.	72
Figura 5.16. Resultados obtenidos del clustering espectral con 2 clusters	76
Figura 5.17. Resultado de la agrupación en 3 clusters para la segunda fase.....	77
Figura 5.18. Resultado de la agrupación en 3 clusters para la segunda fase.....	77
Figura 5.19. Resultado de la agrupación en 5 clusters para la tercera fase.....	78
Figura 5.20. Resultado de la agrupación en 2 clusters para la tercera fase.....	78
Figura 5.21. Resultado final del clustering espectral con acierto del 94.74%.....	79
Figura 5.22. Resultado obtenido mediante agrupación en 10 clusters. HOG 4 ventanas y 16 bins.....	80
Figura 5.23. Resultado obtenido mediante agrupación en 15 clusters. HOG 4 ventanas y 16 bins.....	81
Figura 5.24. Resultado obtenido mediante agrupación en 20 clusters. Gist 8 ventanas y 8 máscaras.....	81
Figura 5.25. Resultado obtenido mediante agrupación en 25 clusters. Gist 8 ventanas y 16 máscaras.....	82
Figura 5.26. Resultado obtenido mediante agrupación en 30 clusters. Gist 8 ventanas y 8 máscaras.....	83

Índice de Gráficas

Gráfica 1. Comparativa de tiempos empleando la Firma de Fourier	35
Gráfica 2. Comparativa de tiempos empleando HOG	36
Gráfica 3. Comparativa de tiempos empleando Gist	36
Gráfica 4. Comparativa de tiempos entre los tres descriptores	37
Dendrogramas obtenidos empleando la Firma de Fourier con el conjunto reducido de imágenes	39
Dendrogramas obtenidos empleando HOG con el conjunto reducido de imágenes.....	40
Dendrogramas obtenidos empleando Gist con el conjunto reducido de imágenes.....	43
Resultados obtenidos empleando como descriptor la Firma de Fourier con el conjunto reducido de imágenes. Clustering jerárquico.....	45
Resultados obtenidos empleando como descriptor HOG con el conjunto reducido de imágenes. Clustering jerárquico.....	46
Resultados obtenidos empleando como descriptor Gist con el conjunto reducido de imágenes. Clustering jerárquico.....	47
Resultados obtenidos empleando la Firma de Fourier para k=4. Clustering jerárquico.....	51
Resultados obtenidos empleando la Firma de Fourier para k=8. Clustering jerárquico.....	51
Resultados obtenidos empleando la Firma de Fourier para k=16. Clustering jerárquico.....	52
Resultados obtenidos empleando la Firma de Fourier para k=32. Clustering jerárquico.....	52
Resultados obtenidos empleando la Firma de Fourier para k=64. Clustering jerárquico.....	53
Dendrograma distancia Coseno - método Ward. Clustering jerárquico.....	53
Dendrograma distancia Spearman - método Average. Clustering jerárquico.....	54
Dendrograma distancia Cityblock - método Average. Clustering jerárquico.....	54
Resultados obtenidos empleando HOG para 1 ventana y 4 bins. Clustering jerárquico.....	55
Resultados obtenidos empleando HOG para 1 ventana y 8 bins. Clustering jerárquico.....	55
Resultados obtenidos empleando HOG para 1 ventana y 16 bins. Clustering jerárquico.....	55
Resultados obtenidos empleando HOG para 2 ventanas y 4 bins. Clustering jerárquico.....	56

Resultados obtenidos empleando HOG para 2 ventanas y 8 bins. Clustering jerárquico.....	56
Resultados obtenidos empleando HOG para 2 ventanas y 16 bins. Clustering jerárquico.....	57
Resultados obtenidos empleando HOG para 4 ventanas y 4 bins. Clustering jerárquico.....	57
Resultados obtenidos empleando HOG para 4 ventanas y 8 bins. Clustering jerárquico.....	57
Resultados obtenidos empleando HOG para 4 ventanas y 16 bins. Clustering jerárquico.....	58
Resultados obtenidos empleando HOG para 8 ventanas y 4 bins. Clustering jerárquico.....	58
Resultados obtenidos empleando HOG para 8 ventanas y 8 bins. Clustering jerárquico.....	59
Resultados obtenidos empleando HOG para 8 ventanas y 16 bins. Clustering jerárquico.....	59
Dendrograma distancia Cityblock - método Average. Clustering jerárquico.....	60
Dendrograma distancia Spearman - método Weighted. Clustering jerárquico.....	60
Dendrograma distancia Correlation - método Average. Clustering jerárquico.....	61
Resultados obtenidos empleando Gist para 1 ventana y 4 máscaras. Clustering jerárquico..	61
Resultados obtenidos empleando Gist para 1 ventana y 8 máscaras. Clustering jerárquico..	62
Resultados obtenidos empleando Gist para 1 ventana y 16 máscaras. Clustering jerárquico..	62
Resultados obtenidos empleando Gist para 2 ventanas y 4 máscaras. Clustering jerárquico..	62
Resultados obtenidos empleando Gist para 2 ventanas y 8 máscaras. Clustering jerárquico..	63
Resultados obtenidos empleando Gist para 2 ventanas y 16 máscaras. Clustering jerárquico..	63
Resultados obtenidos empleando Gist para 4 ventanas y 4 máscaras. Clustering jerárquico..	64
Resultados obtenidos empleando Gist para 4 ventanas y 8 máscaras. Clustering jerárquico..	64
Resultados obtenidos empleando Gist para 4 ventanas y 16 máscaras. Clustering jerárquico..	64
Resultados obtenidos empleando Gist para 8 ventanas y 4 máscaras. Clustering jerárquico..	65
Resultados obtenidos empleando Gist para 8 ventanas y 8 máscaras. Clustering jerárquico..	65
Resultados obtenidos empleando Gist para 8 ventanas y 16 máscaras. Clustering jerárquico..	65
Dendrograma distancia Correlation - método Complete. Clustering jerárquico.....	66
Dendrograma distancia Spearman - método Centroid. Clustering jerárquico.....	67
Resultados obtenidos empleando como descriptor la Firma de Fourier. Clustering espectral..	73
Resultados obtenidos empleando como descriptor HOG. Clustering espectral.....	74
Resultados obtenidos empleando como descriptor Gist con el conjunto reducido de imágenes. Clustering espectral.....	75

Índice de Tablas

Tabla 1. Tipos de distancia posibles de emplear en la función <i>pdist</i>	21
Tabla 2. Métodos de medición de distancia posibles de emplear en la función <i>linkage</i>	22
Tabla 3. Matriz de porcentajes de acierto	26
Tabla 4. Matriz de asignación. Paso 1	26
Tabla 5. Matriz asignación. Paso 2	27
Tabla 6. Matriz de asignación. Paso 3	27



Capítulo 1.

Introducción

1.1. Robótica móvil

Un robot móvil es un vehículo autónomo que se desplaza por un entorno mientras realiza una tarea.

La **navegación autónoma** se puede definir como la capacidad de moverse de forma segura a lo largo de una trayectoria entre un punto de inicio y uno final evitando los obstáculos que se presenten. Para resolver el problema de la navegación autónoma se deben seguir los siguientes pasos:

- 1º) Conocer dónde se encuentra el robot. Localización
- 2º) Saber dónde se quiere acudir. Creación de mapas del entorno o mapping
- 3º) Definir la ruta para ir de un punto inicial a un punto final. Planificación de trayectorias
- 4º) Navegar siguiendo la trayectoria definida. Seguimiento de trayectorias

El primer paso, la localización, tiene como objetivo tratar de estimar la posición en la que se encuentra el robot respecto a las coordenadas globales del mundo. Para ello, el robot, obtiene información mediante sensores internos, como puede ser la odometría, y de sensores externos, entre los que se encuentran sensores láser, sensores de visión... [2].

El siguiente paso consiste en construir una representación útil del entorno, es decir, un mapa. Este mapa sustentará las labores de localización posteriores y la planificación de trayectorias.

Una vez conocido el mapa del entorno, se debe definir la trayectoria a seguir por el robot. La ruta establecida deberá evitar todo tipo de obstáculos.

Finalmente, se calculan las velocidades para que el robot se desplace siguiendo la trayectoria definida. Durante el seguimiento de la ruta, se debe asegurar el reconocimiento del entorno en cada momento para evitar posibles obstáculos.

Hoy en día, y cada vez más, la robótica móvil está presente en múltiples aplicaciones [2], tanto para usos cotidianos como para aplicaciones especiales. Encontramos robots móviles para las tareas de limpieza del hogar, juguetes para niños, así como para asistencia a personas mayores. Por otra parte, este tipo de robots, también se emplea para transporte de mercancías, tareas de vigilancia y rescate, exploración y acceso a lugares peligrosos... En la Figura 1.1 se muestran algunos ejemplos.



Figura 1.1. Diferentes aplicaciones de robots móviles

1.2. Objetivo

El presente trabajo se encuadra dentro de la temática de la creación de mapas y localización de robots móviles.

El punto de partida es un conjunto de escenas omnidireccionales captadas por un robot mientras recorre una ruta a través de un entorno desconocido.

El objetivo que se fija para este trabajo es, empleando únicamente información visual global, tratar de agrupar las escenas para crear mapas topológicos.

Para ello se estudiarán y se analizarán diferentes técnicas de descripción de apariencia global y diferentes algoritmos de clustering.

1.3. Contenidos

El trabajo presenta los siguientes contenidos:

El **capítulo 2**, se dedica a los descriptores de apariencia global. Comienza con una visión general acerca de los descriptores, sus ventajas e inconvenientes y sus principales propiedades. A continuación, se describen las diferentes técnicas de extracción de la información visual que se van a estudiar, la Firma de Fourier, el Histograma de Orientación del Gradiente y Gist. Se mostrará el fundamento de cada uno de los algoritmos así como los parámetros a configurar de cara a mejorar los resultados para la tarea de creación de mapas.

A continuación, en el **capítulo 3**, se dan a conocer nociones básicas de la creación de mapas. En primer lugar, se explican los diferentes tipos de mapas que podemos encontrar en función de las características de su representación. Después, se enumeran las diferentes etapas en la creación de mapas y a continuación se plantea cómo emplear cada uno de los métodos de descripción de escenas en esta tarea. Finalmente, se proponen dos métodos diferentes a la hora de crear estos mapas.

El **capítulo 4**, se dedica al agrupamiento o clustering de imágenes. La primera parte de este capítulo profundiza en el clustering de tipo jerárquico y se detalla el funcionamiento y la configuración de las funciones empleadas en éste. En la segunda parte del capítulo se muestra el algoritmo de clustering espectral así como las funciones empleadas en éste.

Una vez estudiados todos los algoritmos necesarios, en el **capítulo 5**, se analizan los resultados obtenidos con cada uno de los descriptores de apariencia visual global explicados anteriormente. En primer lugar se da a conocer la base de datos empleada así como el robot y el sistema de visión con el que han sido capturadas las imágenes. Seguido de esto se explica la adaptación que se realiza de las imágenes antes de procesarlas y se estudia el coste computacional para cada descriptor. A continuación se analizan los resultados obtenidos para un conjunto reducido de escenas el cual permite comprobar de una forma más sencilla y reducida la eficacia de los métodos. Finalmente se analizan los resultados obtenidos de procesar el conjunto total de imágenes.

Por último, en el **capítulo 6**, se recogen las conclusiones finales del estudio y se proponen futuras líneas de investigación.



Capítulo 2.

Descriptores de Apariencia Visual Global

Un aspecto necesario para dotar a los robots de autonomía es el hecho de que sean capaces de crear mapas y localizarse en ellos.

En primer lugar, se debe captar información visual del entorno mediante sensores de visión para posteriormente procesarla.

Para ello es necesario extraer la información necesaria de cada una de las imágenes captadas por el sensor utilizado para ser capaces de representar el entorno en el que se encuentra el robot de una manera robusta. De esta manera se podrá estimar su posición y orientación y por tanto, el robot podrá desplazarse por el entorno.

El uso de descriptores de apariencia global de escenas es una técnica que permite extraer esta información sin tener que recurrir a las landmarks que dificultan este proceso para entornos dinámicos [3].

La función principal de este método es representar cada imagen utilizando un descriptor, un vector que contiene las principales características que permiten describir la imagen. Tratan la imagen en su totalidad, sin extraer ninguna información local en concreto.

Las dos principales ventajas que ofrecen los descriptores de apariencia global son la reducción del coste computacional, frente a trabajar con las imágenes directamente, y la eliminación de información redundante o irrelevante [4].

Los procesos de creación de mapas y localización se llevan a cabo mediante la comparación entre los descriptores de las diferentes imágenes.

Existen diferentes algoritmos de descripción para trabajar con descriptores de apariencia global. En este trabajo profundizaremos en la Firma de Fourier, el Histograma de Orientación del Gradiente y Gist.

La correcta elección de los parámetros del descriptor empleado es determinante para llegar a un compromiso entre la precisión y el coste computacional. Sobre todo este último requisito es muy importante a la hora de trabajar en tiempo real. En el capítulo 5 analizaremos los resultados obtenidos con las diferentes configuraciones de los parámetros.

2.1. Propiedades de los descriptores

Para que un descriptor sea eficiente, debe contar con las siguientes propiedades [3]:

- El descriptor debe contener la información principal de la escena en una dimensión reducida, es decir, debe ejercer el efecto de compresión de la información.
- Debe existir una relación de correspondencia entre la distancia entre descriptores y la distancia geométrica entre los puntos en los que se capturan las imágenes.
- Los descriptores deben ser robustos frente a situaciones en las que pueda ocasionarse cambios de luminosidad, oclusiones, ruido...
- El coste computacional debe ser lo suficientemente bajo como para trabajar en tiempo real en el problema de localización.
- Es necesario que los descriptores cuenten con información de la orientación en la que se encontraba el robot en el momento de capturar la imagen.

A continuación se describen los métodos de descripción global de imágenes que se van a emplear en este trabajo así como sus propiedades y los parámetros a configurar para cada uno de ellos.

2.2. Descriptores basados en la Transformada Discreta de Fourier.

La Transformada Discreta de Fourier de una imagen se define como:

$$I(u, v) = \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} i(x, y) \cdot h(x, y) \cdot e^{-j2\pi(ux+vy)} = A(u, v) \cdot e^{j\Phi(u,v)}$$

donde $i(x, y)$ es la distribución de intensidad de la imagen respecto de las variables espaciales (x, y) , u y v son las variables de frecuencia espacial (ciclos/pixel) y $h(x, y)$ es una ventana para reducir los efectos de las discontinuidad de principio y final de la imagen. La función compleja resultante, $I(u, v)$ es la Transformada de Fourier de la imagen, y puede ser descompuesta en dos términos reales, el espectro de amplitud de la imagen $A(u, v) = |I(u, v)|$ y el argumento de la transformada de Fourier, $\Phi(u, v)$, el cual contiene información sobre la forma y la posición de los componentes de la imagen.

El término de la amplitud ofrece información acerca de la estructura de la imagen. Representa la distribución de frecuencias espaciales a lo largo de la imagen. De ésta se puede

obtener información de orientación, suavidad, longitud y anchura de los contornos de la imagen. El cuadrado del módulo de la transformada de Fourier se denomina espectro de energía de la imagen y contiene información de la distribución de la energía de la señal entre las diferentes frecuencias espaciales.

El término del argumento ofrece información de las propiedades locales de la imagen como la forma y la posición de los componentes de la imagen [25], [26].

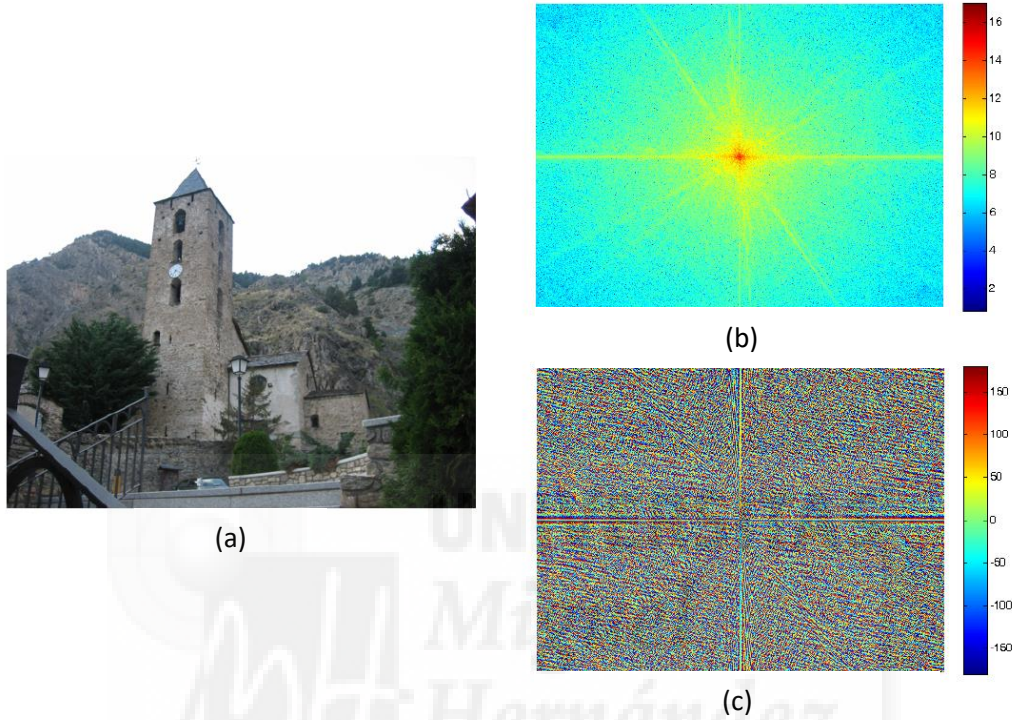


Figura 2.1. Transformada de Fourier 2D de una escena. (a) Imagen original, (b) Módulo de la Transformada de Fourier, (c) Fase de la Transformada de Fourier

2.2.1. Firma de Fourier

La firma de Fourier es un método basado en la Transformada directa de Fourier para la descripción de escenas panorámicas [5]. Presenta invariancia ante rotación en el plano del movimiento y posee un bajo coste computacional.

Al partir de una imagen panorámica $im_j(x, y) \in \mathbb{R}^{N_x \times N_y}$ se puede extraer la información más relevante de la imagen por medio de la transformada discreta de Fourier de sus filas. Para calcular la Firma de Fourier, es necesario expandir cada fila de la imagen panorámica $\{a_n\} = \{a_0, a_1, \dots, a_{N_y-1}\}$ usando la DFT 1D en la secuencia de números complejos $\{A_n\} = \{A_0, A_1, \dots, A_{N_y-1}\}$.

De este modo se obtendrá una nueva matriz $d_j(u, y) \in \mathbb{C}^{N_x \times N_y}$ del mismo tamaño que la original pero que contendrá la mayor parte de la información en los componentes de baja frecuencia de cada fila. Esto permitirá trabajar con las k_1 primeras columnas de la matriz obtenida. Esta matriz, llamada Firma de Fourier, puede ser descompuesta en dos términos, la matriz de módulos $A_j(u, y) = |d_j(u, y)|$ y la matriz de argumentos $\phi_j(u, y)$ con N_x filas y k_1 columnas.

Dos imágenes que hayan sido tomadas en el mismo punto pero con distinta orientación, tendrán los mismos valores en las filas correspondientes pero desplazados un determinado número de columnas que dependerá de la variación en la orientación del robot. Por tanto, el módulo de ambas imágenes será igual a pesar de que la orientación cambie.

Esto nos permite realizar el cálculo de la posición del robot y el cálculo de la orientación por separado. La matriz de módulos permite estimar la posición del robot mientras que la matriz de argumentos puede utilizarse para estimar la orientación del robot respecto a una orientación de referencia.

Este método permite describir cada imagen de manera independiente al resto, sin necesidad de conocer ninguna información de las demás, lo que favorece el trabajo en tiempo real.

2.3. Descriptores basados en Histogramas de Orientación del Gradiente

Este método es utilizado tradicionalmente en la detección de objetos y personas en escenas [6], [7]. Es un método de fácil construcción y que cuenta con un coste computacional asequible.

La apariencia de los objetos de una imagen y su forma pueden ser descritas mediante la distribución de la intensidad de los gradientes y la dirección de los bordes.

Para implementar este descriptor, se divide la imagen en pequeñas regiones conectadas conocidas con el nombre de celdas. Para cada una de estas celdas se calcula un histograma de orientación del gradiente. Posteriormente estos histogramas son normalizados calculando una medida de intensidad dentro de una región más grande llamada bloque. Esta normalización mejora la robustez del descriptor ante cambios en la iluminación.

Los pasos a seguir en el cálculo de este descriptor se mencionan a continuación.

1º) *Cálculo del gradiente*. En este paso se aplican dos máscaras, una en orientación vertical y otra en horizontal para extraer el gradiente y se realiza la convolución entre ambas. A continuación se calcula la magnitud y la orientación del gradiente píxel a píxel.

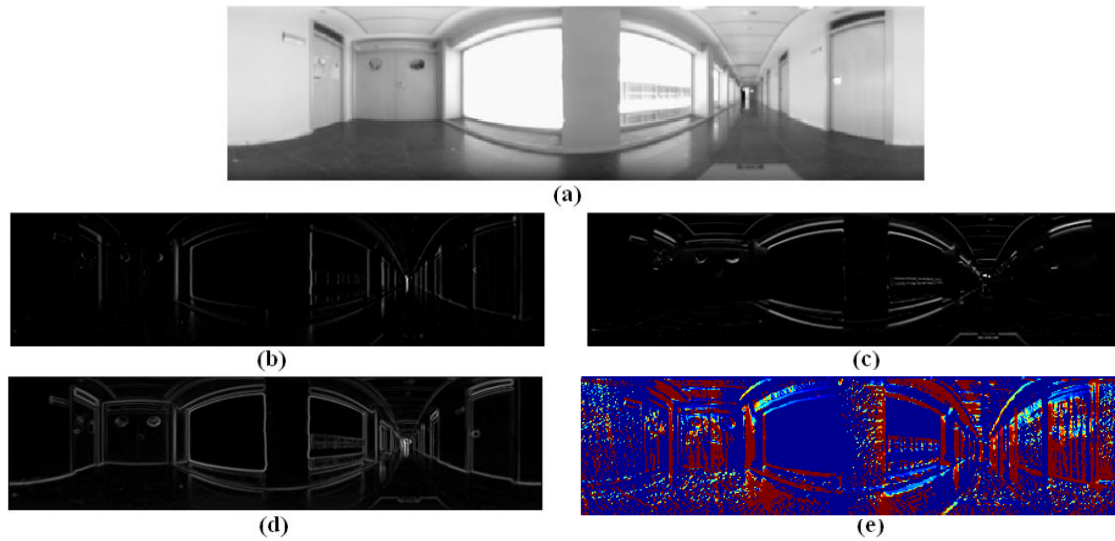


Figura 2.2. (a) Imagen panorámica inicial, (b) componente horizontal, (c) componente vertical, (d) módulo del gradiente y (e) fase del gradiente.

2º) *Clasificación de las orientaciones.* Consiste en dividir la imagen en celdas y crear el histograma de orientación de cada una de ellas. Los canales del histograma se distribuyen uniformemente en el intervalo de 0° a 180° . Cada píxel ponderará con un factor igual a la magnitud del gradiente en el canal correspondiente a su orientación. El motivo por el que la imagen se divide en celdas horizontales es para que el resultado sea invariante a rotación. En una ventana horizontal de dos imágenes tomadas en el mismo punto pero con diferente orientación el único cambio que notaremos es que los elementos aparecen desplazados.

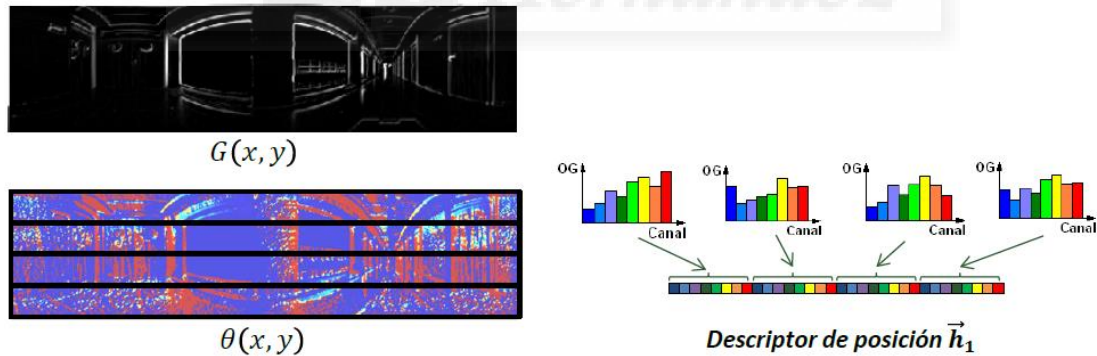


Figura 2.3. Histograma de orientación

3º) *Bloques del descriptor.* Para poder tener en cuenta cambios de iluminación, es necesario normalizar localmente los gradientes. Para ello, las celdas se agrupan en bloques más grandes. El descriptor será el vector que contiene los componentes de los histogramas de las celdas, normalizados a partir de la información de los bloques a los que pertenece cada una de las celdas.

4º) *Normalización de los bloques*. Se normalizan los bloques empleando una de las normas estándar.

2.4. Descriptores basados en Gist

Este tipo de descriptor se basa en imitar la habilidad del sistema de percepción humano para reconocer una escena a través de la identificación de determinadas regiones que destacan por ser de un color o textura diferente en comparación con el entorno. El concepto de Gist fue empleado por primera vez bajo el nombre de *holistic representation of the spatial envelope* por Oliva et al [8], [9].

El objetivo que se persigue es obtener un descriptor que permita localizar la escena ante aliasing visual y que sea invariante a rotación.

Los pasos a seguir para construir este descriptor son los siguientes [10]:

1º) *Construcción de una pirámide de imágenes*. Se construye una pirámide de imágenes Gaussiana de modo que el primer nivel es la imagen original y para obtener cada nuevo nivel se aplica progresivamente un filtro paso bajo. Cada imagen resultante se muestrea para reducir su resolución.



Figura 2.4. Pirámide de imágenes Gaussiana

2º) *Filtrado de Gabor*. Este filtrado permite incluir en el descriptor la información referente a la orientación de cada imagen de los cuatro primeros niveles de la pirámide anterior. Se realiza un filtrado mediante 4 filtros de Gabor de diferente orientación, obteniéndose 4 matrices para cada nivel de la pirámide.

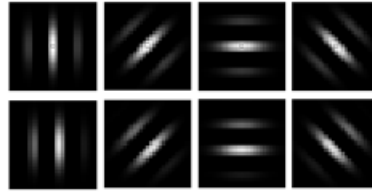


Figura 2.5. Filtros de Gabor

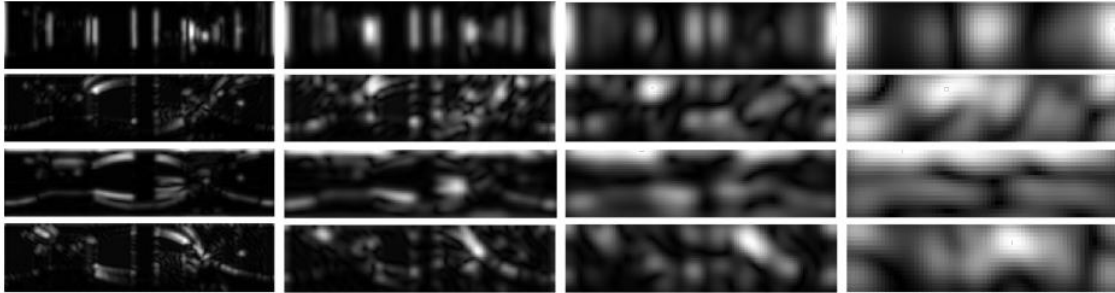


Figura 2.6. Filtrado de la imagen mediante filtros de Gabor

3º) *Reducción de la dimensión*. Dado que cada imagen se ha convertido en 16 matrices con características de orientación, es necesario reducir su dimensión. Para ello se agrupan los píxeles en bloques y posteriormente se reduce la información mediante el Análisis de Componentes Principales que permite trabajar con un número de componentes menor. En la Figura 2.7 se han agrupado los píxeles en bloques. Para el desarrollo de este trabajo, los píxeles serán agrupados en ventanas horizontales para que, al igual que con HOG, consigamos un descriptor invariante a rotación.

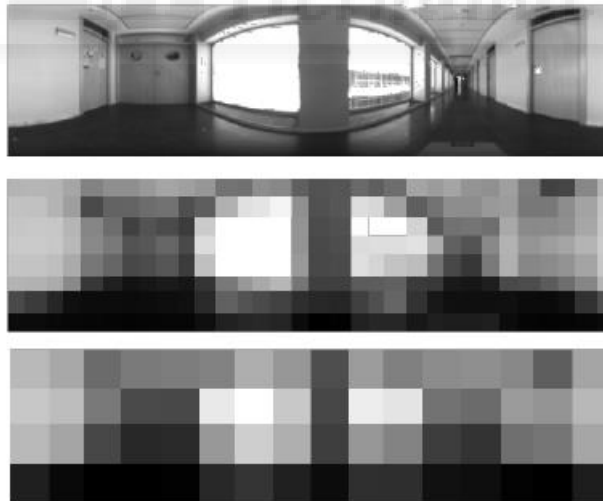


Figura 2.7. Agrupación de píxeles en bloques para reducir la dimensión de la imagen



Capítulo 3.

Creación de mapas

3.1. Tipos de mapas

La creación de mapas es un problema que presenta cierta incertidumbre debido a la información de que se dispone acerca del entorno real. A la hora de crear los mapas empleados para la navegación de los robots móviles, podemos encontrarnos con mapas de 3 tipos: mapas métricos, mapas topológicos y mapas híbridos.

- Los mapas **métricos** son representaciones basadas en coordenadas. Expresan relaciones espaciales entre entidades básicas. Este tipo de mapas proporciona las coordenadas de cada objeto respecto al sistema de referencia global basándose en la precisión geométrica.

Podemos encontrar diferentes tipos:

- *Representación de ocupación*. El espacio se divide en celdas y para cada una de ellas se indica si está ocupada o libre [12], [13].



Figura 3.1. Representación de ocupación

- *Representaciones geométricas.* El modelo consta de un conjunto de objetos geométricos parametrizados como puntos, rectas, curvas... que describen las fronteras del espacio libre respecto a un sistema de referencia global.

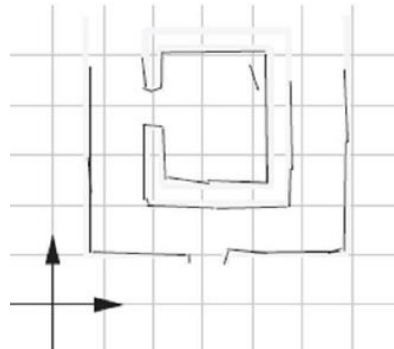


Figura 3.2. Representación geométrica

- *Representaciones basadas en landmarks.* El entorno se representa a partir de un conjunto de objetos destacados o landmarks, los cuales están especificados en un sistema de referencia global.

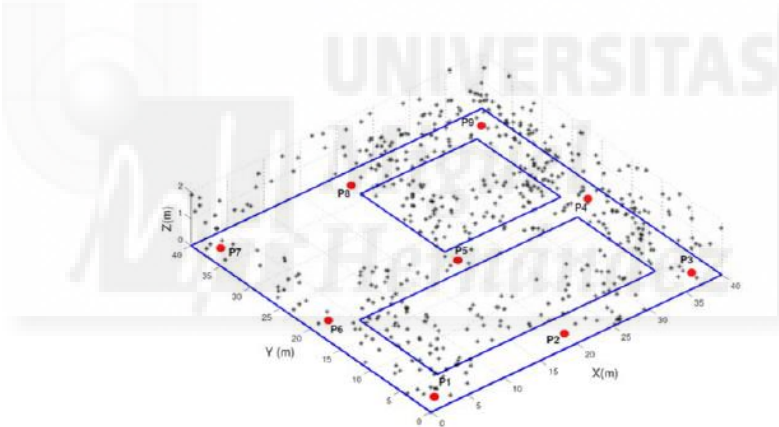


Figura 3.3. Representación mediante landmarks

- Los mapas **topológicos** son representaciones relacionales, es decir, expresan relaciones espaciales entre las entidades básicas de forma explícita [15]. Este tipo de mapas muestra la relación que existe entre las diferentes entidades.

Encontramos diferentes tipos:

- *Grafos de vistas.* Constan de nodos y enlaces entre ellos. Los nodos son las localizaciones y los enlaces entre ellos indican la adyacencia espacial entre las diferentes localizaciones.



Figura 3.4. Grafo de vistas

- *Grafos tipo ruta.* Representaciones en la que los nodos corresponden a lugares característicos del entorno y los enlaces corresponden a los caminos que conectan las localizaciones y que por tanto permiten navegar de un lugar a otro [14].

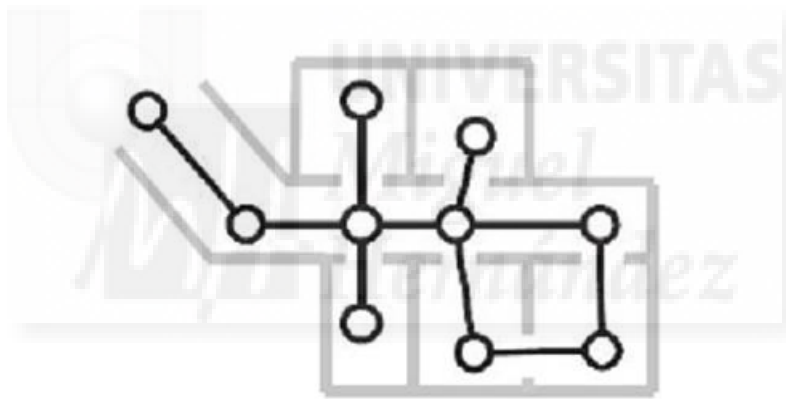


Figura 3.5. Grafo tipo ruta

- Los mapas **híbridos** son una combinación de los dos anteriores [17]. La información topológica permite crear un mapa de alto nivel y la información métrica el mapa un mapa preciso de bajo nivel. En el mapa de alto nivel se representan zonas amplias con características visuales similares y las relaciones de conectividad entre ellas. En el mapa de bajo nivel se representa el contenido de los nodos anteriores de forma métrica con mayor nivel de detalle.

Los diferentes tipos que encontramos son:

- *Representación por capas.* Cada entorno está representado por varias capas y cada una de estas capas emplea su propia representación y cubre el entorno completo.

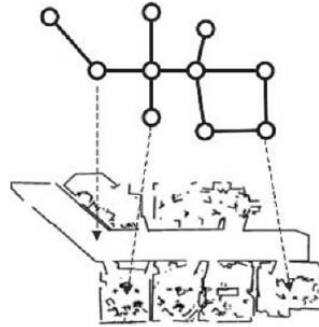


Figura 3.6. Representación por capas

- *Organización jerárquica.* Consta también de varias capas que cubren el entorno completo pero todas ellas emplean el mismo modo de representación en distintos niveles de detalle [16].

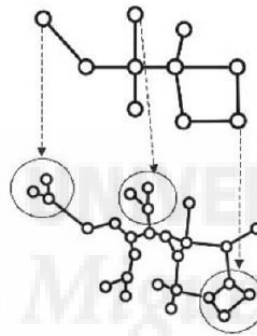


Figura 3.7. Organización jerárquica

- *Patchworks.* Se representan subregiones del entorno individualmente. El tipo de representación es común pero cada subregión utiliza su propio sistema de referencia. La relación entre ellas se basa en una representación de mayor nivel que constituye una representación lineal.

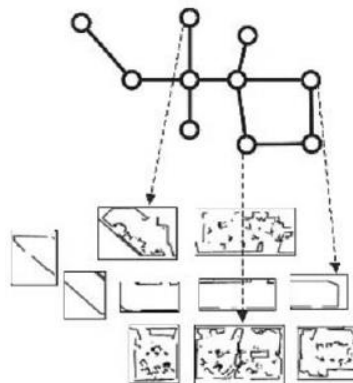


Figura 3.8. Patchwork

Este trabajo se dedica a la construcción de mapas de tipo topológico mediante algoritmos de clustering jerárquico y espectral, es decir, mapas que expresan relaciones espaciales en distintos niveles de detalle.

3.2. Etapas en la creación de mapas y localización

El problema de creación de mapas y localización se puede subdividir en 2 etapas, aprendizaje y validación [3].

La etapa de **aprendizaje** consiste en crear el modelo del entorno. Mediante el sensor visual, en nuestro caso una cámara omnidireccional, se toman las imágenes y posteriormente se describen a partir de los descriptores. Para construir el mapa, se hace uso de las relaciones entre los descriptores.

Posteriormente, en la etapa de **validación**, el robot toma una imagen de una posición desconocida, se calcula su descriptor y se compara con el resto de descriptores de la etapa anterior. El resultado de esta comparación nos permite obtener la posición y la orientación del robot.

Este trabajo se destina principalmente a la primera etapa, la etapa de aprendizaje.

3.3. Creación de mapas topológicos

Como se detallaba anteriormente, el estudio profundiza en la construcción de mapas topológicos, ya que se dispone únicamente de información de las relaciones de conectividad entre las escenas pero no de datos métricos, y se analizan los resultados para la fase de aprendizaje.

3.3.1. Creación de mapas empleando la Firma de Fourier

Al utilizar el descriptor de apariencia global basado en la Firma de Fourier, obtenemos una representación de cada una de las imágenes en forma de descriptor. A la hora de configurar el algoritmo, se variará el parámetro k que es el número de columnas de la matriz de módulos que retenemos.

Los valores para k con los que trabajaremos son: 4, 8, 16, 32 y 64.

3.3.2. Creación de mapas empleando HOG

Para crear el mapa empleando el método de Histograma de Orientación del Gradiente, los parámetros a configurar serán el número de celdas o ventanas horizontales y el número de bins o canales en cada histograma.

El número de ventanas horizontales con los que trabajaremos serán 1, 2, 4 y 8 y para cada una de ellos analizaremos los resultados con 4, 8 y 16 bins.

3.3.3. Creación de mapas empleando Gist

Cuando empleamos el método basado en Gist, se crea una pirámide de imágenes a la que se le aplican un número de máscaras o filtros de Gabor que será uno de los parámetros a estudiar en la configuración del método.

Posteriormente, para reducir la cantidad de información, los píxeles de las matrices resultantes del apartado anterior se agrupan en bloques similares a las celdas o ventanas horizontales del descriptor HOG, este parámetro también será configurable.

El número de filtros de Gabor que se aplicarán serán 4, 8 y 16 y el número de ventanas horizontales 1, 2, 4 y 8.

3.4. Métodos para la creación de mapas topológicos

El objetivo final que se persigue en este trabajo es obtener un mapa topológico basándose en la información visual de las escenas capturadas por el robot.

Para construir el mapa topológico se pueden seguir dos métodos [11]:

1. Creación del mapa en dos fases: clustering jerárquico + clustering espectral
2. Creación del mapa en una única fase: clustering espectral.

En el primero de los métodos, se emplea en primer lugar el algoritmo de clustering jerárquico para realizar una primera clasificación, de manera que los clusters obtenidos se correspondan con las habitaciones o estancias del entorno. El mapa obtenido se considera un mapa de alto nivel. En este primer paso juegan un papel importante los descriptores de apariencia visual y la correcta configuración de los parámetros de las funciones que intervienen en el proceso de clustering jerárquico.

En segundo lugar, se emplea el algoritmo de clustering espectral para que, de cada uno de los clusters obtenidos en el paso anterior, se realice un nuevo agrupamiento. Como se ha explicado anteriormente, el algoritmo de clustering espectral empleado se basa en contrastar la similitud entre descriptores. El mapa obtenido en este paso se considera un mapa de nivel intermedio.

Capítulo 4.

Agrupamiento o clustering de imágenes

Una vez obtenidos los descriptores de apariencia visual global de las escenas, el siguiente paso es comenzar con la clasificación de las imágenes.

El proceso de clustering consiste en agrupar las escenas haciendo uso de los descriptores según sus características. Según Everitt et al [21] los clusters son regiones continuas del espacio I -dimensional que contienen una alta densidad de puntos, separadas de otras regiones con alta densidad de puntos por zonas con una relativamente baja densidad de puntos.

El objetivo perseguido es obtener un mapa jerárquico, es decir, que agrupe la información en varios niveles.

La tarea de clustering se puede dividir en los siguientes pasos [11]:

1º) *Selección de características*. En este paso entran en juego los descriptores de apariencia global. Como se comentaba en capítulo 2, el objetivo principal de éstos es extraer la mayor cantidad de información de la manera más comprimida posible.

2º) *Elección de la medida de distancia*. La elección de la medida de distancia puede influir directamente en la precisión del resultado de clustering.

3º) *Selección del algoritmo de clustering y configuración de los parámetros*. El siguiente paso consiste en seleccionar el algoritmo con el que se va a realizar el agrupamiento y configurar los parámetros según el número de clusters que se desea obtener.

4º) *Verificación de los resultados*. Tras obtener los resultados del agrupamiento, se ha de comprobar la precisión y la eficacia del algoritmo empleado.

Los algoritmos de agrupamiento empleados en el presente trabajo son el algoritmo de clustering jerárquico y el algoritmo de clustering espectral.

4.1. Clustering jerárquico

El algoritmo de clustering jerárquico agrupa las escenas creando dendrogramas. Un dendrograma es un tipo de representación gráfica en forma de árbol que organiza los datos en categorías que se van dividiendo en otros hasta llegar al nivel de detalle deseado [18].

Los pasos a seguir en el clustering jerárquico son los siguientes:

1º) *Encontrar la similitud o la disimilitud entre cada pareja de escenas.* Para ello se calcula la distancia entre cada par de escenas con la función *pdist*. Esta función permite ser configurada con diferentes tipos de distancias.

2º) *Agrupar las escenas mediante un dendrograma.* Mediante la función *linkage* se agrupan en parejas las escenas más próximas entre sí. Esta función emplea la información de distancia obtenida en el paso anterior. Una vez se han creado las parejas, se van formando clusters agrupando estas parejas ya formadas y, de esta manera, sucesivamente hasta obtener un único cluster, que corresponde al nivel más alto del árbol o dendrograma. Además, mediante la función *dendrogram*, podemos visualizar el dendrograma creado.

3º) *Determinar a qué altura seccionar el dendrograma.* Mediante la función *cluster* se secciona el dendrograma y se asignan los objetos que quedan por debajo de cada corte a un cluster individual. Esto es, el número de ramas que corta la línea horizontal en el dendrograma determina el número de clusters a obtener.

El funcionamiento y configuración de las funciones *pdist*, *linkage*, *dendrogram* y *cluster* se detalla a continuación. Estas funciones que han sido empleadas para la implementación de los métodos son propias del programa Matlab.

4.1.1. Función *pdist*

La función *pdist* calcula la distancia entre cada pareja de escenas. La función permite elegir el tipo de distancia que deseamos utilizar.

$$D = \text{pdist}(X, \text{distance})$$

X: matriz que contiene las escenas

distance: tipo de distancia

D: vector que contiene las distancias calculadas entre cada par de elementos

Los tipos de distancia con los que vamos a trabajar y analizar los resultados son los que se muestran en la Tabla 1.

Distancia	Descripción
Euclidean	Distancia por defecto. $d_{st}^2 = (x_s - x_t)(x_s - x_t)'$
Cityblock	$d_{st} = \sum_{j=1}^n x_{sj} - x_{tj} $
Minkowski	$d_{st} = \sqrt[p]{\sum_{j=1}^n x_{sj} - x_{tj} ^p}$
Chebychev	$d_{st} = \max_j \{ x_{sj} - x_{tj} \}$
Cosine	$d_{st} = 1 - \frac{x_s x_t'}{\sqrt{(x_s x_s')(x_t x_t')}}}$
Correlation	$d_{st} = 1 - \frac{(x_s - \bar{x}_s)(x_t - \bar{x}_t)'}{\sqrt{(x_s - \bar{x}_s)(x_s - \bar{x}_s)'} \sqrt{(x_t - \bar{x}_t)(x_t - \bar{x}_t)'}}$
Spearman	$d_{st} = 1 - \frac{(r_s - \bar{r}_s)(r_t - \bar{r}_t)'}{\sqrt{(r_s - \bar{r}_s)(r_s - \bar{r}_s)'} \sqrt{(r_t - \bar{r}_t)(r_t - \bar{r}_t)'}}$

Tabla 1. Tipos de distancia posibles de emplear en la función *pdist*

4.1.2. Función linkage

Una vez calculada la distancia entre los diferentes elementos, calculamos mediante la función *linkage* cuáles de ellos deberían ser agrupados en clusters. Esta función crea un árbol jerárquico a partir del vector de distancias obtenido con la función *pdist*.

$$Z = \text{linkage}(D, \text{method})$$

D: vector de distancias obtenido con la función *pdist*.

method: indica cómo medir la distancia entre clusters.

Z: matriz que contiene las parejas creadas y la distancia entre ellas. En las dos primeras columnas almacena los índices correspondientes a las escenas emparejadas y en la tercera columna almacena la distancia entre las escenas.

La función se puede configurar con los métodos de medición de distancia entre clusters recogidos en la Tabla 2.

Método	Descripción
Average	Distancia media no ponderada $d(r, s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} dist(x_{ri}, x_{sj})$
Weighted	Distancia media ponderada $d(r, s) = \frac{(d(p, s) + d(q, s))}{2}$
Centroid	Distancia al centro de masas sin ponderar $d(r, s) = \ \bar{x}_r - \bar{x}_s\ _2 \quad \text{con} \quad \bar{x}_r = \frac{1}{n_r} \sum_{i=1}^{n_r} x_{ri}$
Median	Distancia al centro de masas ponderada $d(r, s) = \ \tilde{x}_r - \tilde{x}_s\ _2$ x_r y x_s son los centroides ponderados de los clusters r y s . Si el cluster r ha sido creado combinando los clusters p y q , se tiene que: $\tilde{x}_r = \frac{1}{2} (\tilde{x}_p + \tilde{x}_q)$
Single	Distancia más cercana, vecino más cercano $d(r, s) = \min(dist(x_{ri}, x_{sj})), i \in (1, \dots, n_r), j \in (1, \dots, n_s)$
Complete	Distancia más lejana $d(r, s) = \max(dist(x_{ri}, x_{sj})), i \in (1, \dots, n_r), j \in (1, \dots, n_s)$
Ward	Mínima varianza $d(r, s) = \sqrt{\frac{2n_r n_s}{(n_r + n_s)}} \ \bar{x}_r - \bar{x}_s\ _2,$ $\ \cdot\ _2$ es la distancia euclídea x_r y x_s son los centroides de los clusters r y s n_r y n_s son el número de elementos en los clusters r y s

Tabla 2. Métodos de medición de distancia posibles de emplear en la función *linkage*

4.1.3. Función dendrogram

Para manejar de una manera más intuitiva la información, podemos visualizar gráficamente el árbol jerárquico mediante la función *dendrogram*, donde el parámetro Z es la matriz obtenida de la función *linkage*.

dendrogram (Z)

Z : matriz de salida de la función *linkage*

En la Figura 4.1 se muestra un ejemplo de la gráfica obtenida con la función dendrograma para un conjunto de 10 imágenes.

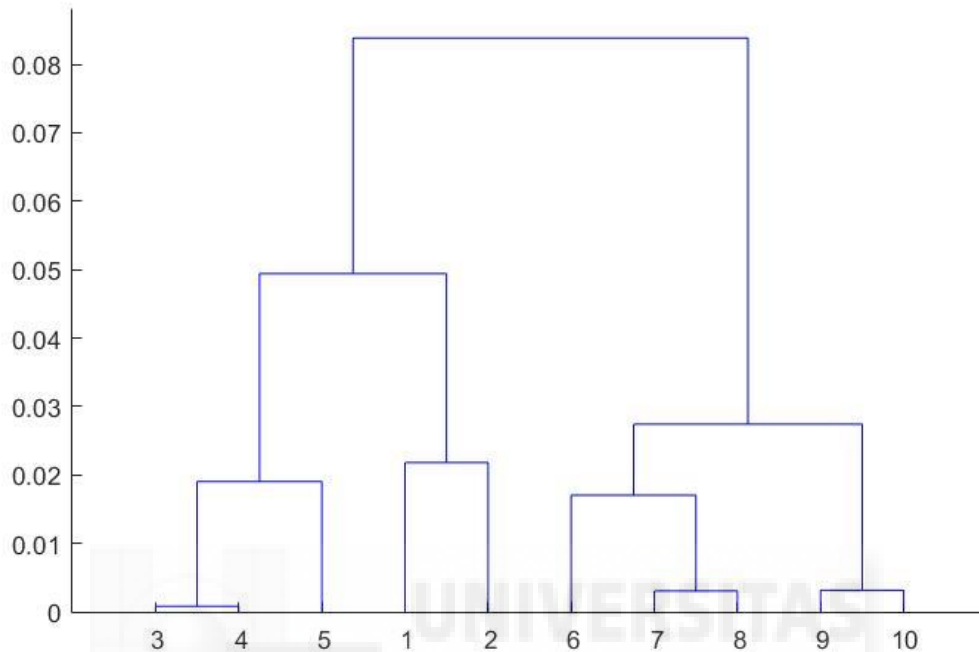


Figura 4.1. Dendrograma obtenido para un conjunto de 10 imágenes

En la gráfica, los números a lo largo del eje horizontal representan los índices de las escenas. Los links entre las escenas están representados por las U invertidas. La altura de las barras verticales indica la distancia entre las imágenes. Un buen agrupamiento será aquel en el que los clusters formados sean considerablemente distantes entre sí. Una forma visual de ver qué elementos quedarían en cada cluster sería trazar una línea horizontal que corte las líneas verticales del dendrograma como se muestra en la Figura 4.2.

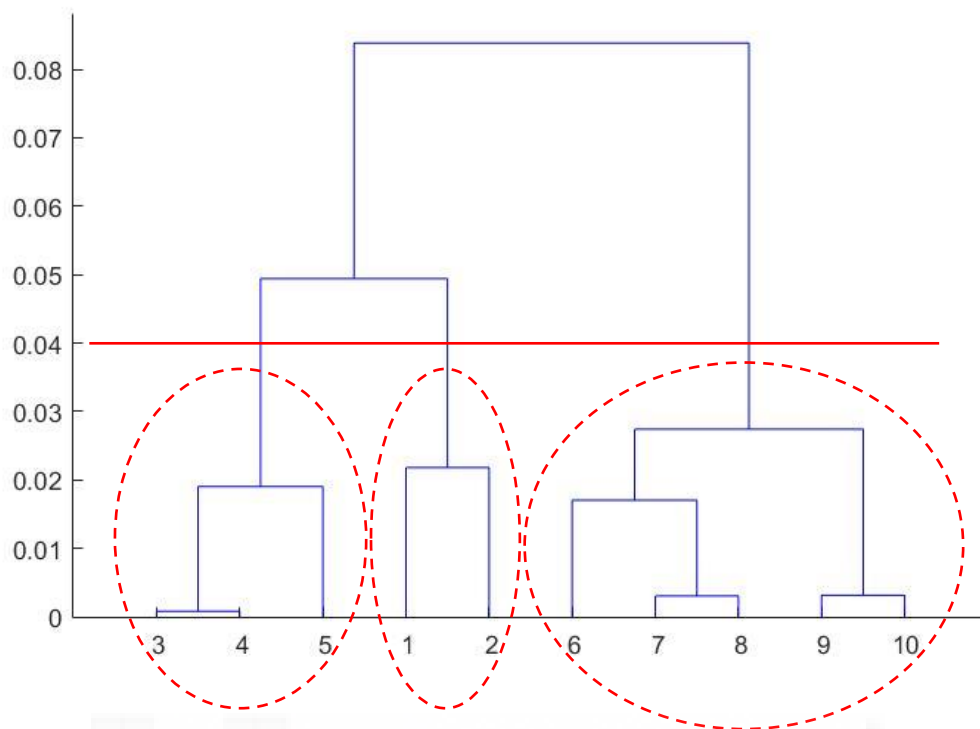


Figura 4.2. Agrupación mediante dendrograma

En este caso, a línea horizontal intersecta tres líneas verticales del dendrograma lo cual quiere decir que estaríamos creando 3 clusters. En el primero de ellos estarían las imágenes 3, 4 y 5, en el segundo estarían las imágenes 1 y 2 y en el tercero estarían la 6, 7, 8, 9 y 10.

4.1.4. Función Cluster

Una vez creado el árbol jerárquico, mediante la función *cluster* se secciona el dendrograma y se asignan los objetos que quedan por debajo de cada corte a un cluster individual.

Esta función permite crear los clusters de dos maneras:

- *División natural de los datos en clusters en función del coeficiente de inconsistencia.* Esta forma no va a ser estudiada en este trabajo dado que las pruebas realizadas no muestran resultados que aporten valor.
- *División especificando el número de clusters que se desea obtener.* La función *cluster* permite indicar como parámetro el número agrupamientos que se quiere crear.

```
T = cluster(Z, 'maxclust', n)
```

Z: matriz de salida de la función *linkage*

maxclust: indicación del modo para crear los clusters

n: número de clusters que se desea obtener

T: etiquetas correspondientes al número de cluster asignado a cada imagen

La función devuelve un vector *T* del tamaño del número de imágenes procesadas que indica a qué cluster se asigna cada una de ellas.

4.1.5. Verificación de precisión

Tras realizar el agrupamiento, se debe comprobar cómo de preciso ha sido este. Para ello, se van a emplear dos métodos en este trabajo.

4.1.5.1. Función *success*

Por una parte, se realiza una comprobación de la precisión mediante una función de propio desarrollo llamada *success*.

```
perc=success(T, c)
```

T: vector de etiquetas obtenido con la función *cluster*

c: número de clusters

perc: porcentaje de acierto

Esta función genera una matriz cuadrada de tamaño igual al número de clusters indicados en la función *cluster*. Basándose en el tamaño que tendrían que tener cada uno de los clusters, cuantifica las veces que se repite cada número en cada cluster y se completa dicha matriz, siendo cada columna una estancia y cada fila un número y dividiendo cada número por el número total de elementos correspondientes a cada estancia. Una vez se tiene la matriz completa, se busca el mayor valor y se asigna para la estancia de esa columna el número correspondiente en su fila. El resto de elementos de esa fila y esa columna se hacen 0. Este proceso se repite hasta que se ha asignado un número para cada estancia. De esta forma se consigue que, aunque el valor que se asigna a cada estancia se haga de forma aleatoria, podamos siempre asignar a cada una el valor que mejores resultados proporcione. A

continuación se muestra un ejemplo para un caso de 3 estancias (A, B y C) que clarifica la explicación.

Vector T obtenido en la función cluster: [2 2 1 2 1 1 3 3 3 3]

Vector ideal de acierto 100%: [A A A B B C C C C C]

Los tres primeros elementos corresponderían a la primera estancia, la estancia A. Seleccionamos los tres primeros elementos del vector T obtenido con la función cluster: [2 2 1]. Cuantificamos el número de veces que se repite cada número en este cluster y rellenamos la primera columna de la tabla. Dividimos el valor por el número total de elementos de la primera estancia, en este caso 3.

Los elementos 4º y 5º pertenecerían a la estancia B. En este caso [2 1]. Repetimos el paso anterior.

El tercer cluster estaría formado por los últimos 5 elementos, [1 3 3 3 3]. Completamos la matriz de la misma forma.

	Estancia A	Estancia B	Estancia C
Nº1	$1/3=0.3$	$1/2=0.5$	$1/5=0.2$
Nº2	$2/3=0.67$	$1/2=0.5$	0
Nº3	0	0	$4/5=0.8$

Tabla 3. Matriz de porcentajes de acierto

Quando tenemos la matriz completa, se asignan los valores a cada estancia. El mayor valor se encuentra en la tercera fila y tercera columna, 0.8. Por tanto a la estancia C se le asigna el valor 3. El resto de elementos de su fila y columna se hacen 0.

	Estancia A	Estancia B	Estancia C
Nº1	$1/3=0.3$	$1/2=0.5$	0
Nº2	$2/3=0.67$	$1/2=0.5$	0
Nº3	0	0	$4/5=0.8$

Tabla 4. Matriz de asignación. Paso 1

Pasamos a buscar el siguiente valor más alto, en este caso 0.6 que se encuentra en la segunda fila y primera columna. Se hacen 0 el resto de valores de su fila y columna. A la estancia A se le asignará el valor 2.

	Estancia A	Estancia B	Estancia C
Nº1	0	1/2=0.5	0
Nº2	2/3=0.67	0	0
Nº3	0	0	4/5=0.8

Tabla 5. Matriz asignación. Paso 2

El último valor que nos queda es 0.5 que se encuentra en la primera fila y segunda columna. A la estancia B se le asignará el valor 1.

	Estancia A	Estancia B	Estancia C
Nº1	0	1/2=0.5	0
Nº2	2/3=0.67	0	0
Nº3	0	0	4/5=0.8

Tabla 6. Matriz de asignación. Paso 3

Por último se ha de computar el porcentaje de acierto total. Para ello se multiplica el porcentaje de acierto de cada estancia por el número de elementos pertenecientes a cada estancia y se divide entre el número total de elementos.

$$0.67 * \frac{3}{10} + 0.5 * \frac{2}{10} + 0.8 * \frac{5}{10} = 0.7 = 70\%$$

El porcentaje de acierto total para este caso ha sido de un 70%.

En este caso, el número de imágenes empleadas es reducido, sin embargo el número de imágenes que se manejan en este trabajo es bastante más elevado. Dado que, al emplear las funciones de clustering, es posible que se asigne la misma etiqueta a imágenes de varias estancias, es necesario emplear una matriz intermedia para impedir que se falseen los resultados. Esto quiere decir, que si se empleara la función *success* tal y como se ha explicado, se podría obtener el 100% de acierto para varias estancias a pesar de que la etiqueta asignada a todas ellas sea la misma.

Esta corrección supone generar una nueva matriz que recoge en cada columna el número de elementos contenidos en cada cluster y multiplicarla por la matriz que contiene el porcentaje de acierto. La búsqueda del mayor valor se realizará entre los elementos de esta nueva matriz pero el valor asignado será el correspondiente en la matriz de porcentaje de acierto.

4.1.5.2. Función NMI

El otro método empleado para medir la precisión en la clasificación es el algoritmo NMI (Normalized Mutual Information). Este método crea una matriz de confusión a partir del vector *T* obtenido en la función *cluster* y el vector ideal de clasificación con el cual se conseguiría una precisión del 100%. A partir de la información de la diagonal principal, que corresponde a las imágenes bien clasificadas, devuelve un índice entre 0 y 1 que indica la precisión de los clusters obtenidos. Cuanto más cercano a 1 sea ese índice, más precisa habrá sido la clasificación.

$$z = nmi(T, y)$$

T: vector obtenido en la función *cluster*

y: vector ideal de precisión 100%

z: índice de precisión

Internamente, esta función, genera la matriz dispersa empleando la función *sparse*.

4.1.6. Verificación de disimilitud

Después de enlazar las escenas en el árbol jerárquico, es recomendable verificar que las distancias, es decir, las alturas de las líneas verticales, reflejan las distancias originales con precisión. Esta altura se conoce con el nombre de distancia cofenética. Una manera de saber cómo de bien se han agrupado las escenas en el árbol jerárquico es comparando las distancias cofenéticas con la distancia original, que es la obtenida de la función *pdist*. Para ello empleamos la función *cophenet* la cual compara la distancia cofenética y la distancia real y calcula su correlación devolviendo el coeficiente de correlación cofenético. Cuanto más cercano a la unidad sea este coeficiente, más preciso será el clustering.

$$c = cophenet(Z, Y)$$

Z: matriz de salida de la función *linkage*

Y: matriz de salida de la función *pdist*

C: coeficiente de correlación cofenético

4.2. Clustering espectral

Los algoritmos de clustering espectral ofrecen la ventaja de poder trabajar con un elevado número de escenas.

Este algoritmo crea en primer lugar la similitud dos a dos entre todas las escenas y almacena los valores en la llamada matriz de similitud, en la que en cada posición se indica la similitud entre las escenas correspondientes a la fila y columna [19], [20]. Esta técnica hace uso del espectro (autovalores) de la matriz de similitud para llevar a cabo una reducción de la dimensionalidad antes de agrupar los datos con menos dimensiones.

Inicialmente partimos de un conjunto de imágenes que se desean agrupar en un número k de clusters. Se estima la similitud dos a dos entre todos los elementos hasta obtener la matriz de similitudes.

Los pasos a seguir en el clustering espectral son los siguientes [11]:

1º) Calcular la matriz Laplaciana normalizada.

2º) Calcular los k autovectores principales de la matriz Laplaciana y generar una nueva matriz U que organice estos vectores en columnas.

3º) Obtener la matriz T a partir de U normalizando sus filas a la norma 1.

4º) Para $i = 1, \dots, n$ consideramos $\vec{y}_i \in \mathbb{R}^k$ el vector correspondiente a la fila i -ésima de T .

5º) Clusterizar los vectores \vec{y}_i , $i=1, \dots, n$ con el algoritmo k-means o cualquier otro algoritmo de clustering para obtener los clusters.

El hecho de que se tengan en cuenta las similitudes mutuas entre todos los elementos, hace que este tipo de clustering resulte más efectivo que el algoritmo de k-means o el clustering jerárquico.

En cambio, cuando el conjunto de imágenes a tratar es muy elevado, el algoritmo requerirá un tiempo y una memoria considerables para calcular los k autovectores de la matriz Laplaciana. Para evitarlo, en la matriz de similitud, se retienen únicamente los componentes s_{ij} en los que j está entre los t vecinos más cercanos de i o viceversa, siendo t un número relativamente pequeño.

A continuación se detallan las funciones empleadas en el clustering espectral. Estas funciones pertenecen a una librería ajena a las propias de Matlab [27].

4.2.1. Función `gen_nn_distance`

A partir de la matriz de similitud, mediante la función `gen_nn_distance` se crea la matriz dispersa correspondiente, es decir, se hacen cero aquellos elementos de la matriz de similitud que no son relevantes. Este paso permite reducir el tiempo y la memoria necesarios considerablemente.

```
gen_nn_distance (data, n, b, st)
```

data: matriz de descriptores correspondientes a las escenas que se desean agrupar

n: número de vecinos más cercanos

b: tamaño del bloque para partir los datos

st: modo de guardado (0 archivo .mat, 1 archivo .txt, 2 ambos)

La ejecución de la función genera la matriz dispersa correspondiente que es guardada según se indique en el último parámetro de la función.

4.2.2. Función *sc*

Posteriormente, mediante la función *sc*, spectral clustering, se agrupan las imágenes empleando la matriz dispersa obtenida con la función *gen_nn_distance*. La función *sc* se basa en el empleo del algoritmo de las k-medias.

```
[T, t1, t2, t3]=sc(A, s, K)
```

A: Matriz dispersa

s: valor sigma empleado para calcular la similitud. Para s=0 se emplea la técnica de autotuning

K: número de clusters que se desea obtener.

T: etiquetas correspondientes al número de cluster asignado a cada imagen.

t1: tiempo de procesamiento para la descomposición

t2: tiempo de procesamiento para el algoritmo k-means (k-medias)

t3: tiempo total de procesamiento

Capítulo 5.

Análisis de los resultados obtenidos

En el presente capítulo se van a analizar los resultados obtenidos con cada uno de los descriptores de apariencia visual global que se han explicado anteriormente para el problema de construcción de mapas y localización.

En primer lugar se da a conocer la base de datos empleada así como el robot y el sistema de visión con el que han sido capturadas las imágenes. Seguido de esto se explica la adaptación que se realiza de las imágenes antes de procesarlas y se estudia el coste computacional para cada descriptor.

A continuación se analizan los resultados obtenidos para un conjunto reducido de escenas el cual permite comprobar de una forma más sencilla y reducida la eficacia de los métodos.

Finalmente se analizan los resultados obtenidos de procesar el conjunto total de imágenes.

5.1. Base de datos empleada

Las imágenes empleadas para el estudio forman parte de la base de datos COLD [23]. Ésta consiste en tres conjuntos de imágenes tomados en entornos de laboratorios. En nuestro caso hemos empleado imágenes del Language Technology Laboratory del Centro de Investigación Alemán de Inteligencia Artificial en Saarbrücken, Alemania.

Esta base de datos fue especialmente creada para trabajar con algoritmos de localización y reconocimiento de lugares.

El conjunto empleado cuenta con 1064 imágenes omnidireccionales de diferentes estancias de la planta, en este caso una cocina, un baño, una sala de impresoras, un despacho y el pasillo. Para reducir el tiempo de cálculo y trabajar de una manera más cómoda se ha decidido emplear 1 de cada 4 imágenes de la base de datos, siendo un total de 266 imágenes las que se han manejado para el estudio de los algoritmos. El plano en planta que contiene la ruta recorrida se muestra en la Figura 5.1 (ruta en rojo).

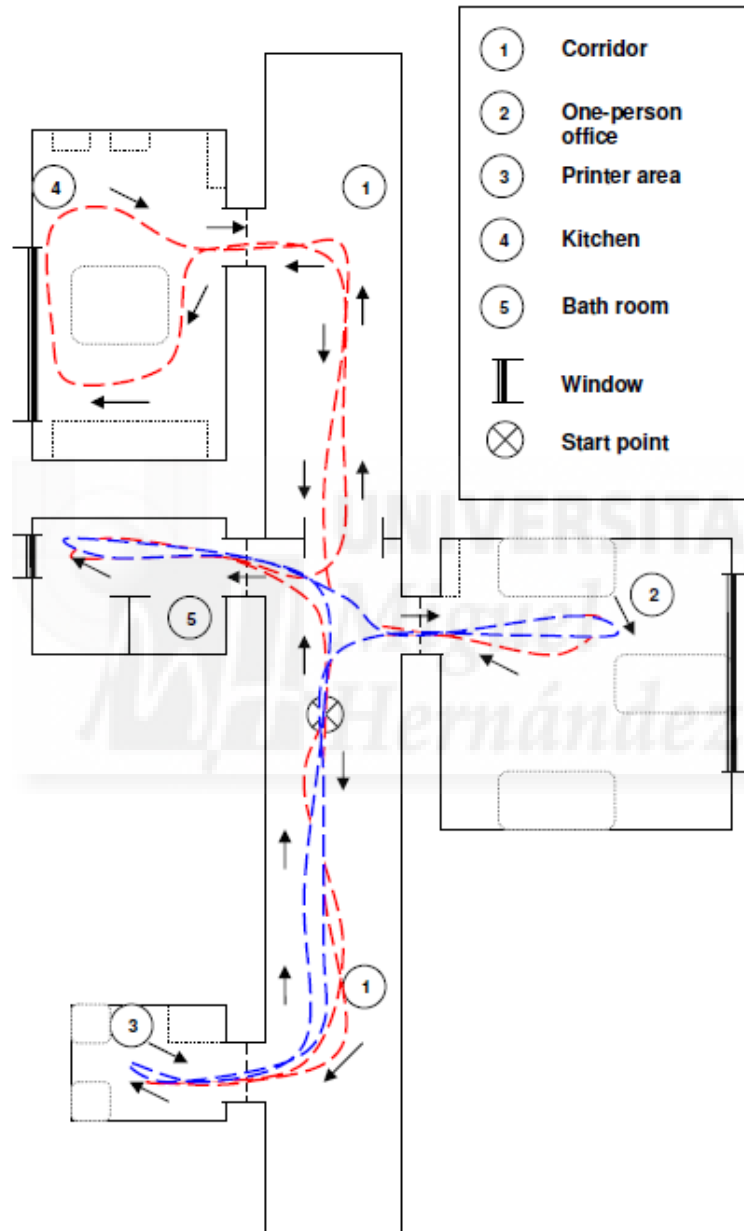


Figura 5.1. Plano en planta de la ruta recorrida. Language Technology Laboratory, Centro de Investigación Alemán de Inteligencia Artificial en Saarbrücken, Alemania

El conjunto de imágenes empleado presenta la dificultad del posible problema de *aliasing visual* que pueda aparecer ya que existen imágenes muy similares visualmente pero que pertenecen a estancias distintas. El fenómeno de *aliasing visual* se da cuando se tienen

imágenes tomadas en puntos alejados entre sí pero presentan una apariencia visual prácticamente idéntica.

5.2. Equipo empleado

El robot móvil empleado para captar las imágenes omnidireccionales fue PeopleBot, equipado con escáner láser SICK y encoders en las ruedas, el cual fue dirigido manualmente mediante el uso de un joystick. Todas las imágenes fueron tomadas en diferentes condiciones de iluminación pero a la misma altura, estando restringido el robot al plano del suelo.

El sistema de las cámaras fue construido utilizando dos cámaras digitales MDCS2 de Videre Design con una resolución de 640×480 píxels; una para las imágenes en perspectiva y otra para las imágenes omnidireccionales. El sistema de visión omnidireccional catadióptrico se construyó utilizando un espejo hiperbólico [22]. Las dos cámaras y el espejo fueron montados en un soporte portátil presentado en la Figura 5.3.



Figura 5.2. Robot móvil PeopleBot



Figura 5.3. Sistema de visión omnidireccional

5.3. Adaptación de las imágenes

El primer paso a la hora de trabajar con el conjunto de imágenes de la base de datos descrita anteriormente es pasar cada una de las imágenes de formato omnidireccional a formato panorámico.

Para ello hemos empleado una función que lee una a una las imágenes y, a partir del conocimiento de la posición del centro de la imagen (coordenadas x , y) y la longitud del radio

exterior e interior, permite transformarlas a formato panorámico. Se ha considerado la misma posición del centro y el mismo radio interior y exterior para todas las imágenes procesadas.



Figura 5.4. Imagen en formato omnidireccional

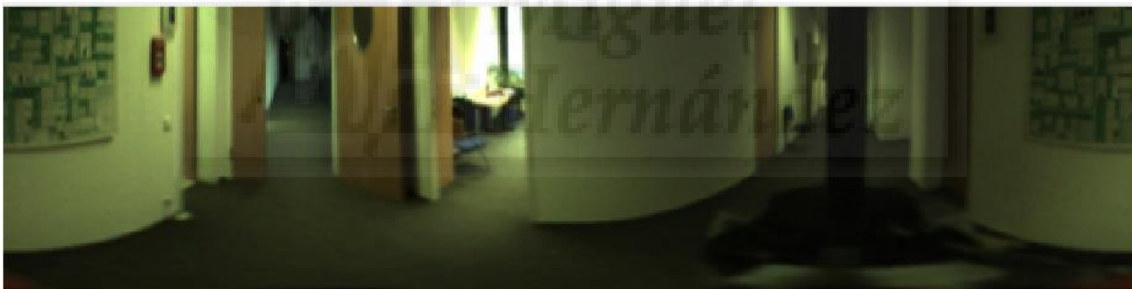


Figura 5.5. Imagen en formato panorámico RGB

Una vez tenemos la imagen en formato panorámico la convertimos de RGB a formato blanco y negro para reducir la cantidad de información dado que las imágenes RGB están formadas por tres capas de intensidad entre 0 y 255 y las imágenes en blanco y negro sólo por una capa. De esta forma el coste computacional se reduce considerablemente.

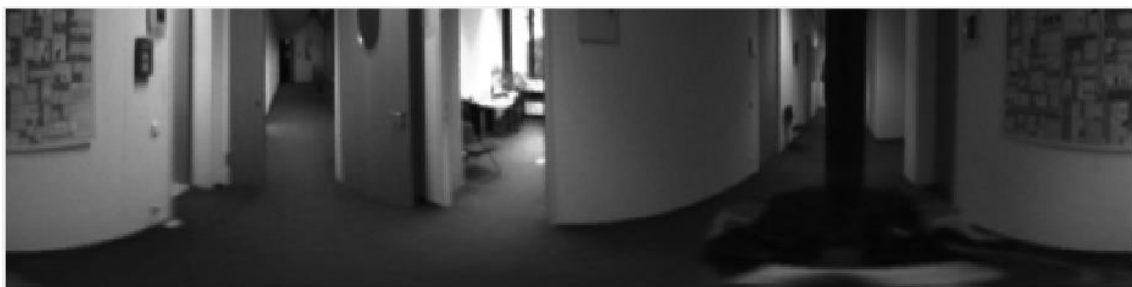


Figura 5.6. Imagen en formato panorámico blanco y negro

La información de cada imagen se almacena en una matriz de 128x512 valores que oscilan entre 0 y 255.

Una vez llegados a este punto, ya se podría comenzar a trabajar con los algoritmos de descripción de apariencia global de las imágenes.

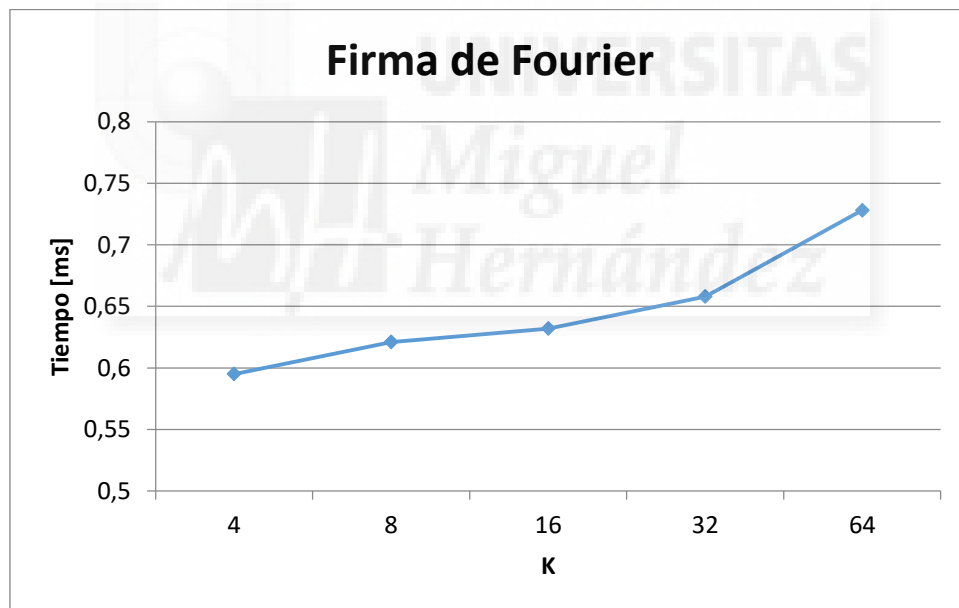
5.4. Estudio del coste computacional

El tiempo de cálculo del descriptor empleado para la creación del mapa, es un factor importante a tener en cuenta ya que será determinante a la hora de trabajar en tiempo real.

Se pretende llegar a un compromiso entre el coste computacional y la precisión del método empleado.

Los valores que se muestran en las gráficas corresponden al tiempo de cálculo del descriptor para una imagen y las unidades son milisegundos.

Para crear el modelo haciendo uso de la Firma de Fourier se obtienen los resultados mostrados en la Gráfica 1:

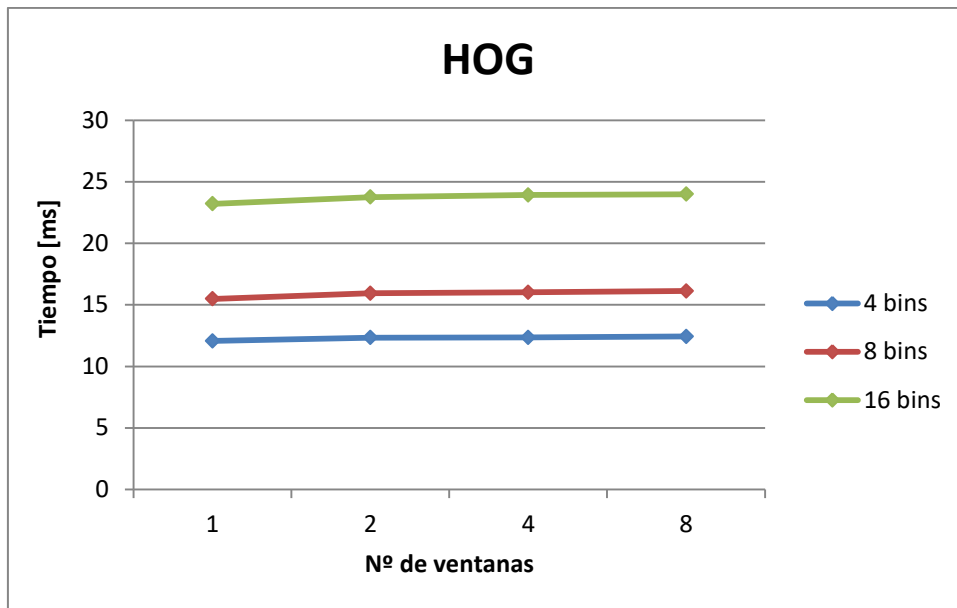


Gráfica 1. Comparativa de tiempos empleando la Firma de Fourier

En el eje de abscisas aparece el número de columnas de la matriz de módulos a retener y en el eje de ordenadas aparece el tiempo que ha supuesto el cálculo.

Como se puede apreciar, se da un crecimiento de tendencia casi lineal, a mayor número de columnas, mayor tiempo de computación. Aun así el incremento es mínimo.

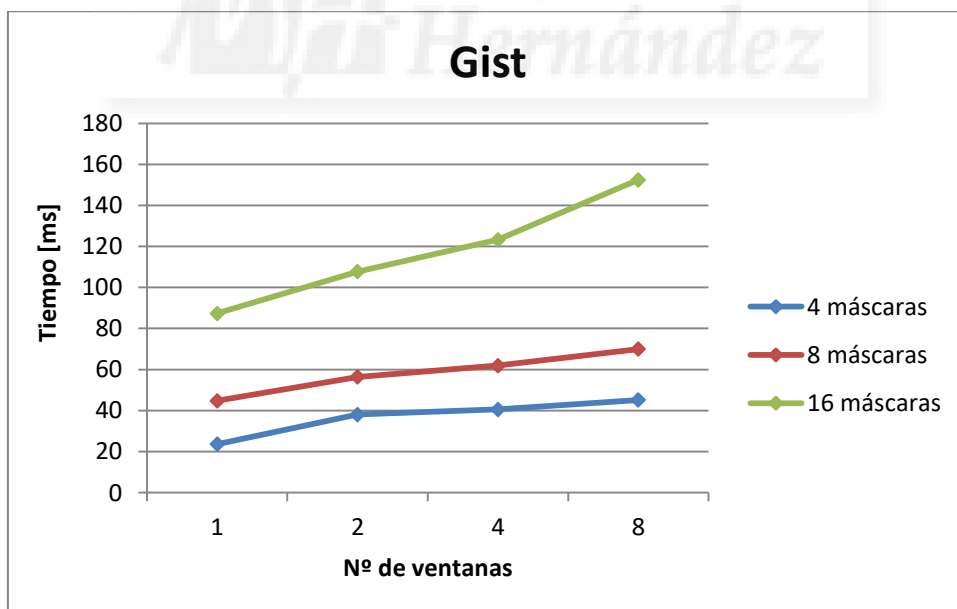
Si hacemos uso del Histograma de Orientación del Gradiente, los resultados obtenidos son los que se muestran en la Gráfica 2.



Gráfica 2. Comparativa de tiempos empleando HOG

Se observa que el aumento del número de ventanas no supone un aumento del tiempo de cálculo. En cambio, el aumento de número de bins o canales de cada histograma sí influye notablemente, mostrando un incremento doble de un salto al siguiente.

Por último, cuando el modelo es creado haciendo uso de Gist se obtienen los tiempos mostrados en la Gráfica 3.



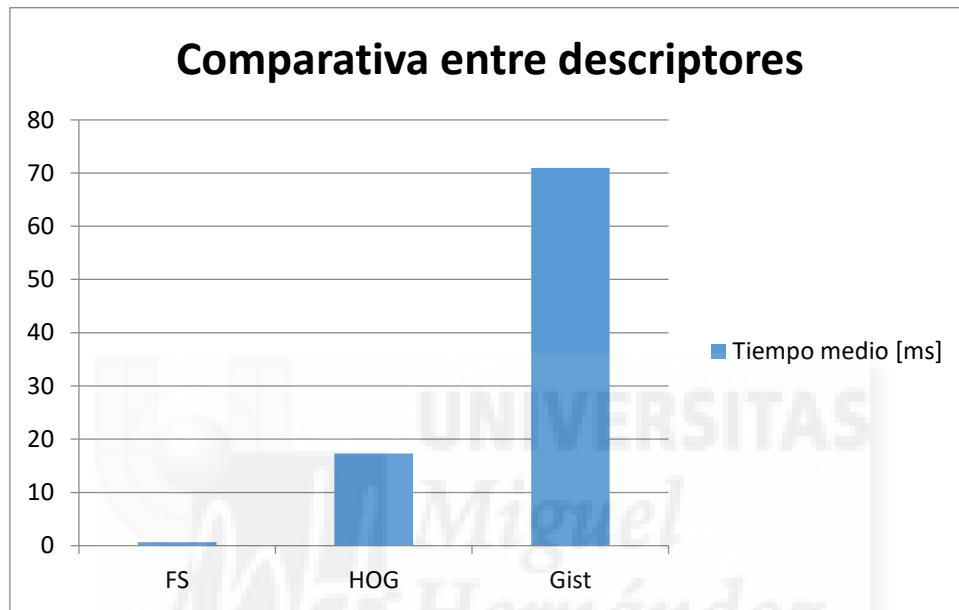
Gráfica 3. Comparativa de tiempos empleando Gist

Se puede observar que tanto el incremento del número de ventanas como el incremento en el número de máscaras suponen un incremento en el tiempo de cálculo. El incremento

debido al aumento del número de ventanas tiene una tendencia lineal mientras que el incremento del número de máscaras supone, al igual que con los bins al emplear HOG, un aumento doble de un salto al siguiente.

La comparación entre los diferentes métodos del tiempo empleado, muestra que el más rápido es la Firma de Fourier, al cual le sigue HOG. En el lugar del más lento se encuentra el método Gist.

La Gráfica 4 muestra el tiempo medio en segundos para cada uno de los métodos:



Gráfica 4. Comparativa de tiempos entre los tres descriptores

5.5. Resultados obtenidos para una selección de escenas

En este apartado se van a analizar los resultados obtenidos para un conjunto reducido de imágenes que han sido seleccionadas a propósito de entre todas las imágenes que conforman la base de datos. Se han elegido 10 imágenes muy diferentes entre sí y de las 5 habitaciones que existentes en la planta fotografiada. El objetivo que se plantea al realizar esta selección es comprobar que los métodos empleados son efectivos y alejarse del posible problema de aliasing visual.

Las imágenes elegidas para este estudio se muestran a continuación.



Baño



Baño

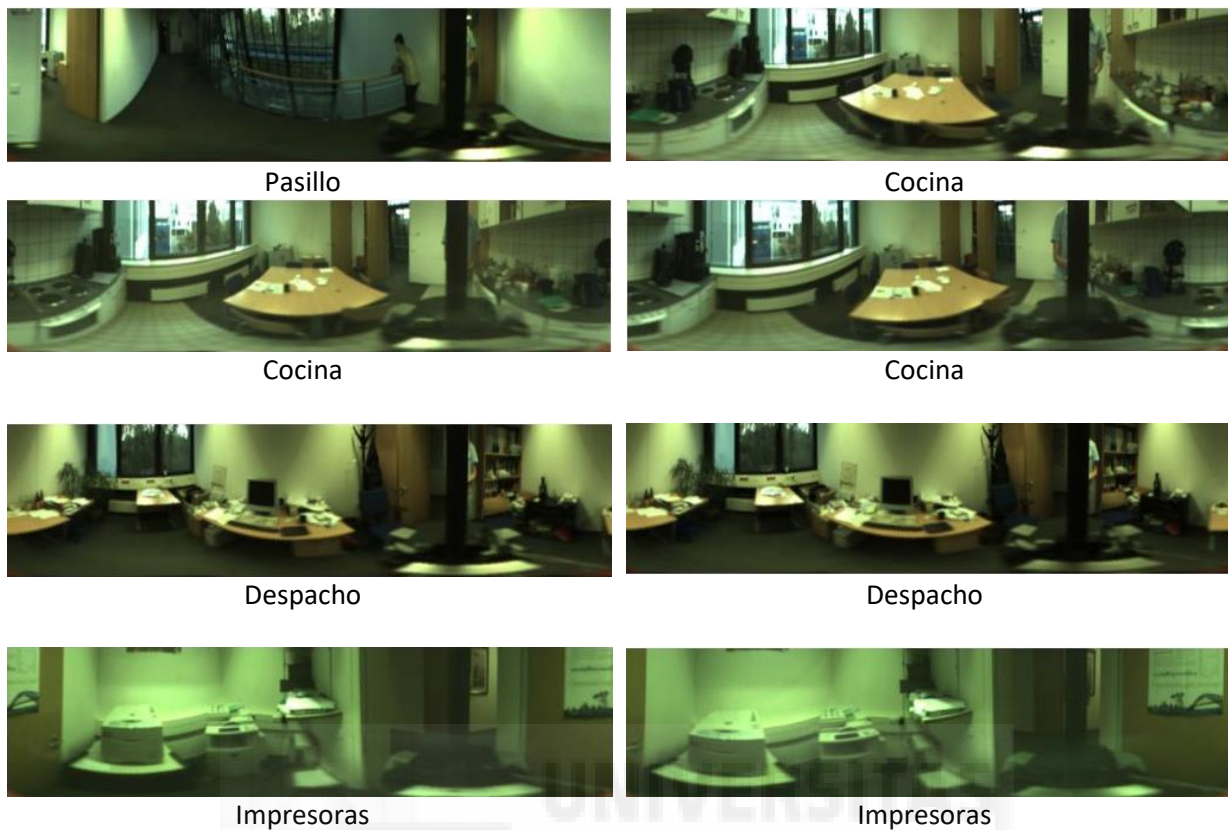


Figura 5.7. Conjunto reducido de imágenes seleccionadas

En primer lugar se calculan los descriptores de apariencia visual global para cada una de las escenas. Como se comentaba en el capítulo 2, los descriptores empleados en este trabajo son la Firma de Fourier, el Histograma de Orientación del Gradiente y Gist.

Para el descriptor basado en la Firma de Fourier, se varía el parámetro k que es el número de columnas de la matriz de módulos que retenemos. Los valores para k con los que trabajamos son 4, 8, 16, 32 y 64.

Para el descriptor basado en el Histograma de Orientación del Gradiente, los parámetros a configurar son el número de celdas o ventanas horizontales y el número de bins o canales en cada histograma. Los números de ventanas horizontales con los que trabajamos son 1, 2, 4 y 8 y para cada una de ellos analizamos los resultados con 4, 8 y 16 bins.

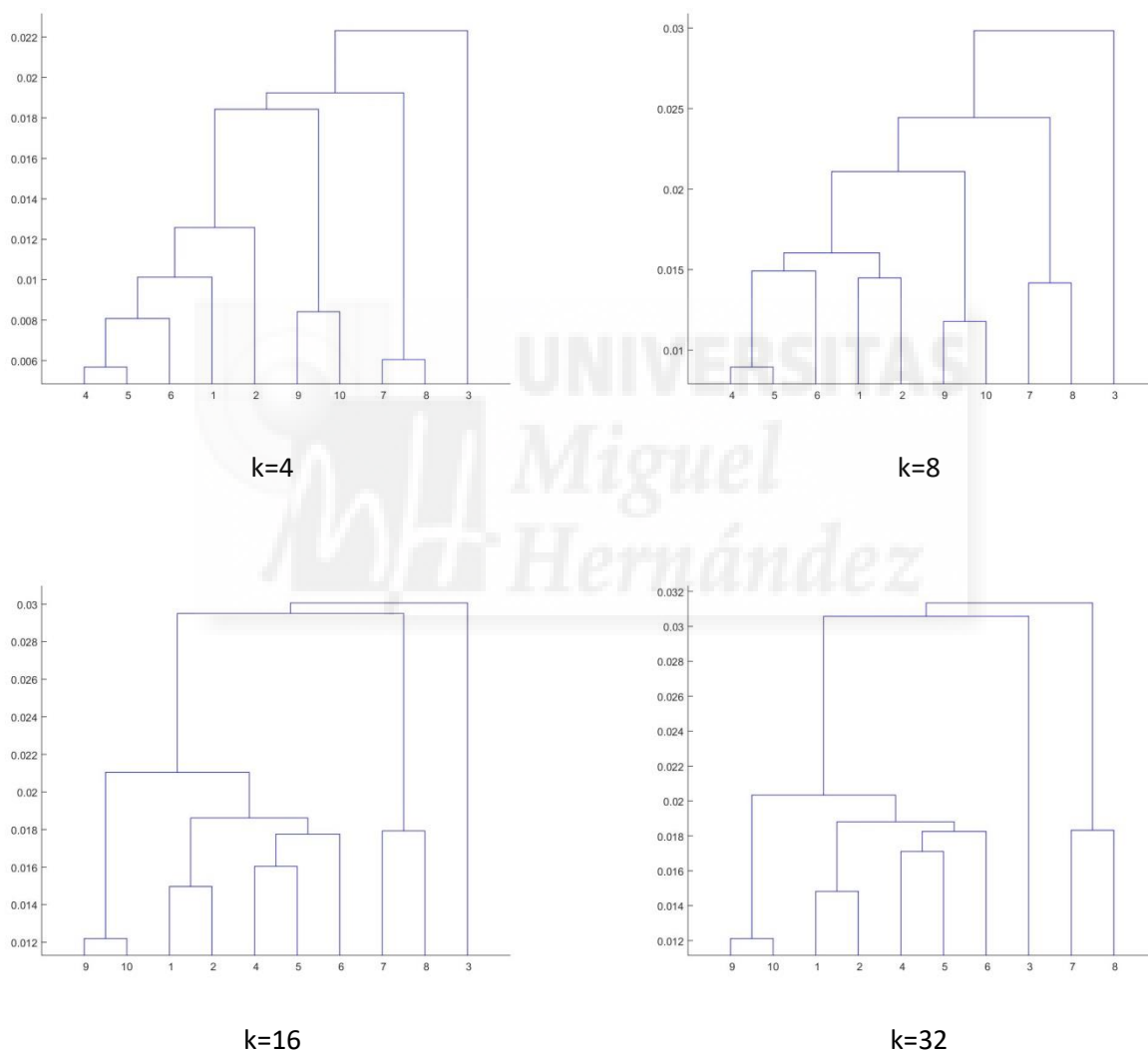
Para el descriptor basado en Gist, los parámetros que se configuran son el número de máscaras o filtros de Gabor a aplicar y el número de ventanas horizontales que se emplean para reducir la información. Los valores con los que trabajamos son 4, 8 y 16 filtros de Gabor y 1, 2, 4 y 8 ventanas horizontales.

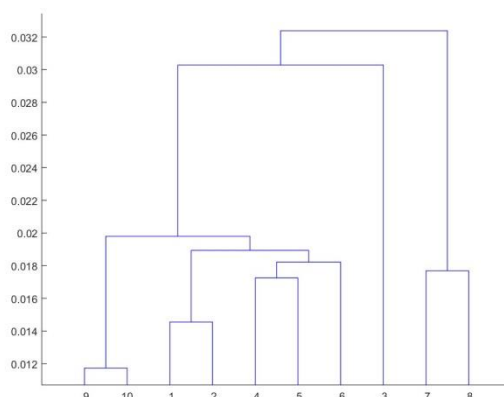
Una vez creados los descriptores, se comienza a realizar la agrupación en clusters. Para encontrar la similitud o la disimilitud entre cada pareja de escenas empleamos la función *pdist*. Para este estudio reducido, se va a emplear únicamente el tipo de distancia *Correlation*.

Después, mediante la función *linkage* se procede a analizar qué escenas deben ser agrupadas para formar los clusters. Para este caso también se van a analizar los resultados únicamente para un método de medición de distancia, *Single*.

El siguiente paso consiste en generar los dendrogramas correspondientes para cada configuración de los parámetros. A continuación se muestran los dendrogramas obtenidos para este conjunto reducido de 10 escenas.

Dendrogramas obtenidos empleando la Firma de Fourier:





k=64

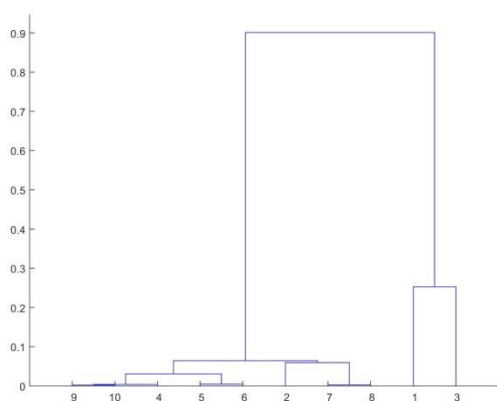
Como se puede apreciar en los dendrogramas, los resultados obtenidos empleado la Firma de Fourier para $k=16$, $k=32$ y $k=64$ son muy similares, mientras que para $k=4$ y $k=8$, los resultados difieren del resto.

En el caso de $k=4$, las imágenes que primero se agrupan son la 4 y la 5, seguido de la 7 y 8. A partir de esta última unión, ya comienzan a aparecer agrupaciones de 3 elementos.

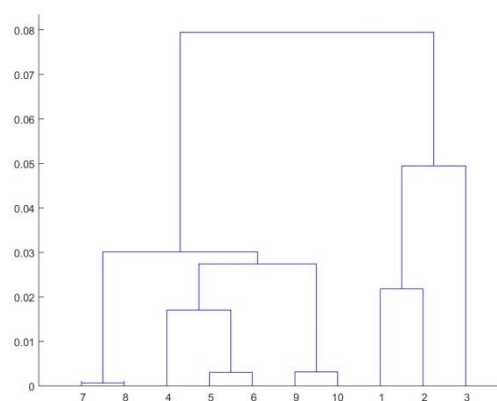
Para $k=8$, se agrupan en primer lugar las imágenes 4 y 5, seguidas de las imágenes 9 y 10, posteriormente 7 y 8 y después 1 y 2. A partir de esta última unión comienzan las agrupaciones de 3 imágenes.

Para $k=16$, $k=32$ y $k=64$, se emparejan en primer lugar las imágenes 9-10, 1-2 y 4-5. A partir de esta última unión ya comienzan los grupos de 3 imágenes. Se puede apreciar, entre estos 3 últimos dendrogramas, una ligera diferencia en la altura de las barras verticales, lo que quedará reflejado en el coeficiente de correlación cofenético.

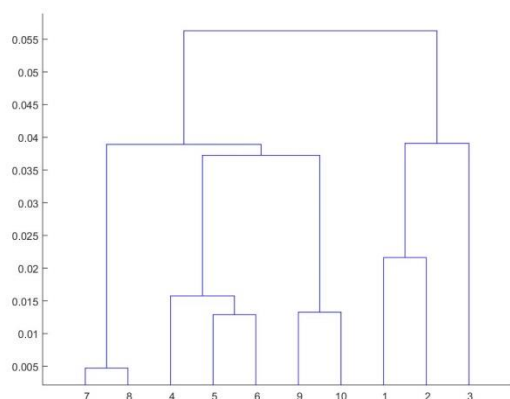
Dendrogramas obtenidos empleando HOG:



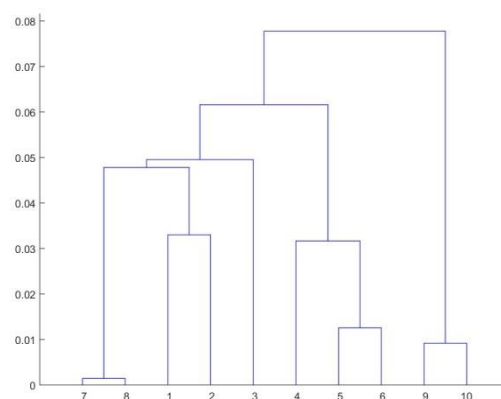
1 ventana y 4 bins



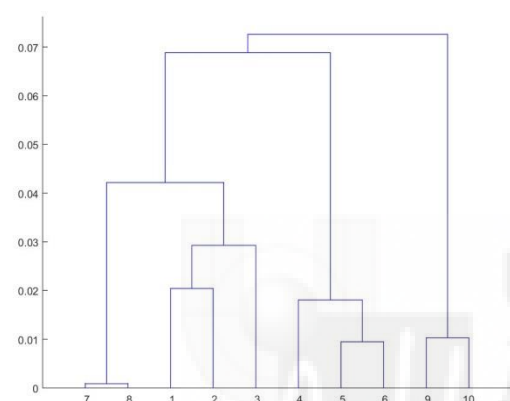
1 ventana y 8 bins



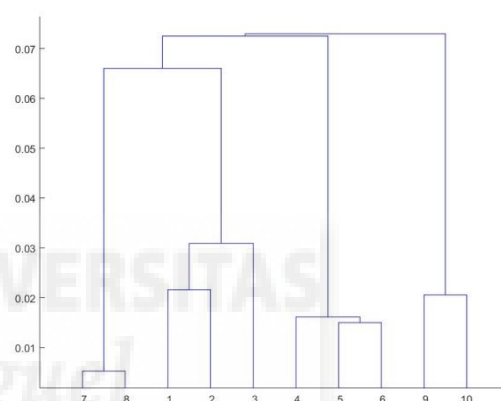
1 ventana y 16 bins



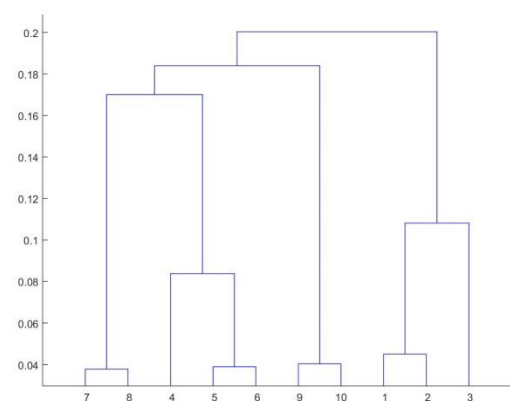
2 ventanas y 4 bins



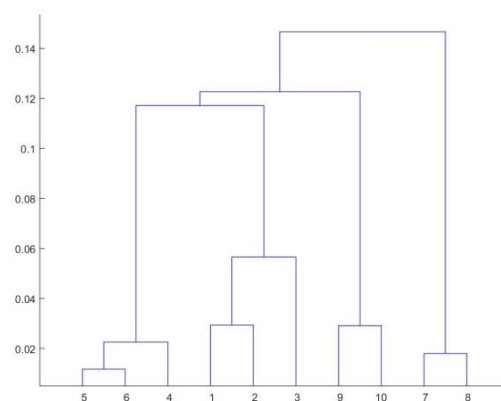
2 ventanas y 8 bins



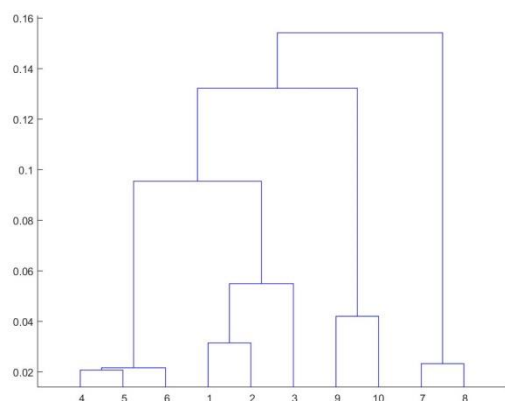
2 ventanas y 16 bins



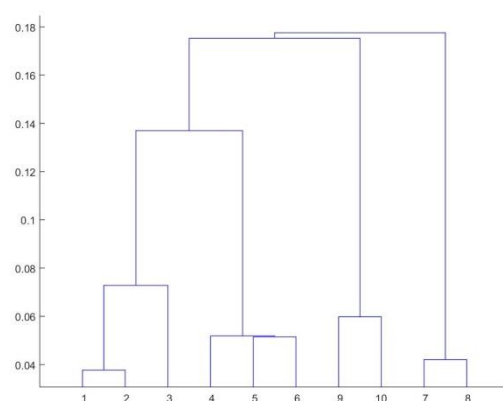
4 ventanas y 4 bins



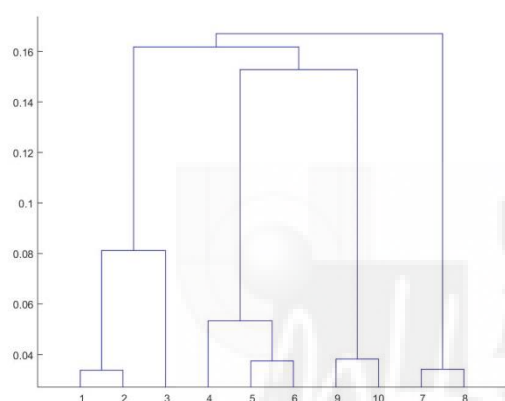
4 ventanas y 8 bins



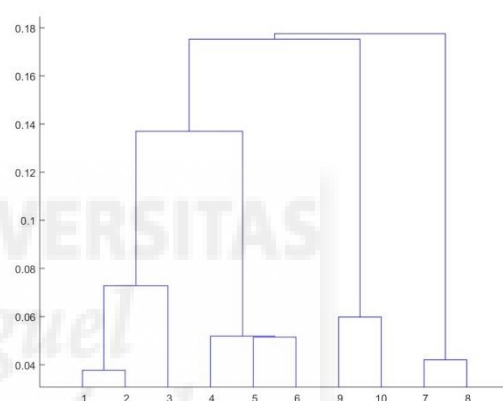
4 ventanas y 16 bins



8 ventanas y 4 bins



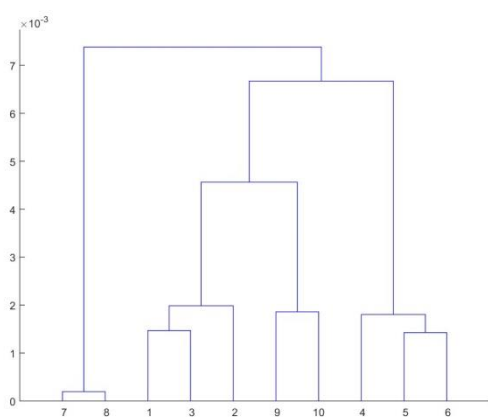
8 ventanas y 8 bins



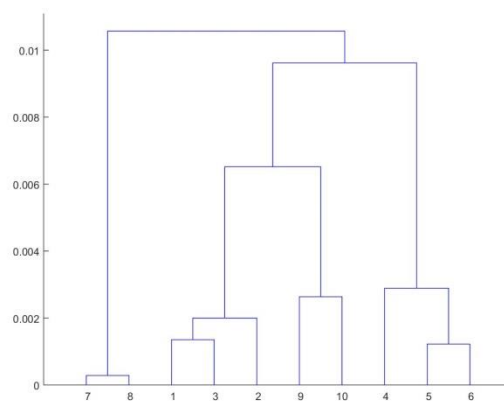
8 ventanas y 16 bins

Para el caso de los dendrogramas obtenidos empleando HOG como descriptor, se puede observar cómo la configuración de los parámetros del descriptor influye en los resultados obtenidos ya que los dendrogramas tienen diferente forma. Aun así, a pesar de que el orden en el que se realizan las agrupaciones cambia para cada configuración, los resultados ofrecidos empleando HOG tienen un alto grado de precisión.

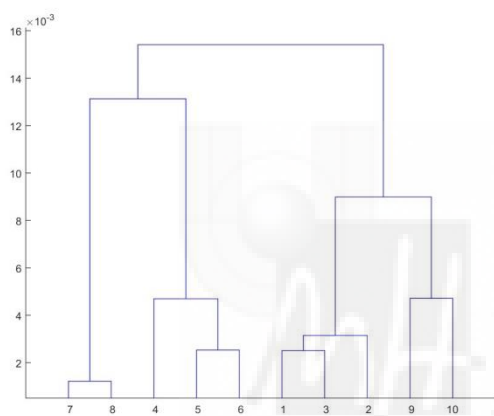
Dendrogramas obtenidos empleando Gist:



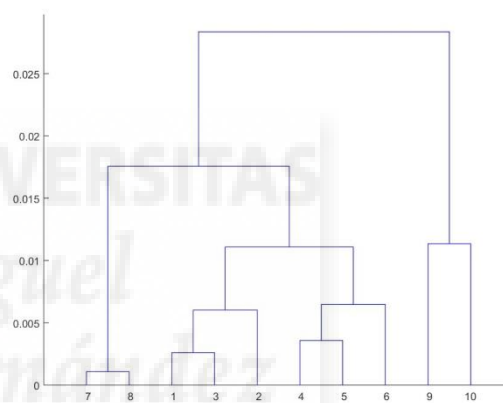
1 ventana y 4 máscaras



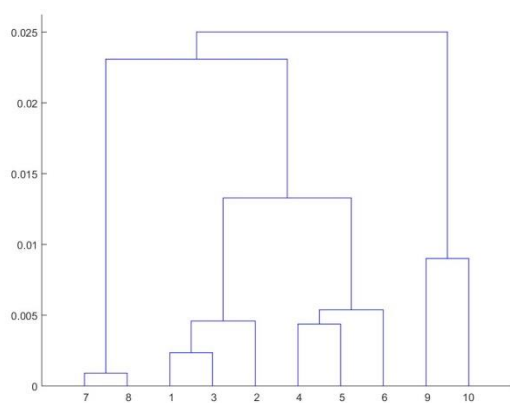
1 ventana y 8 máscaras



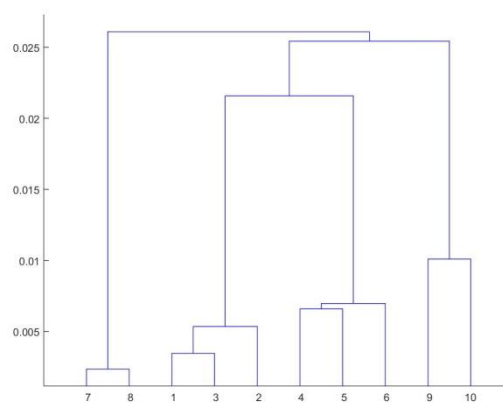
1 ventana y 16 máscaras



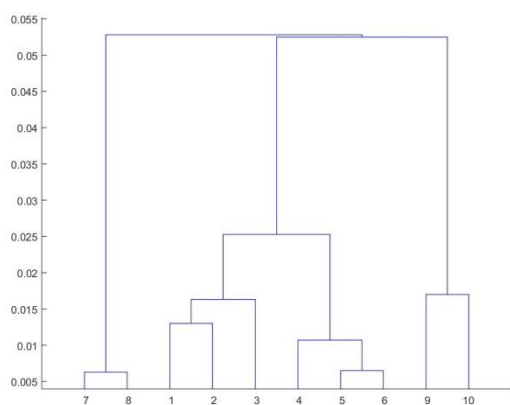
2 ventanas y 4 máscaras



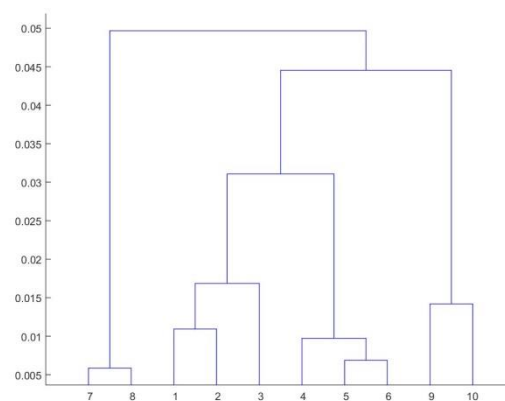
2 ventanas y 8 máscaras



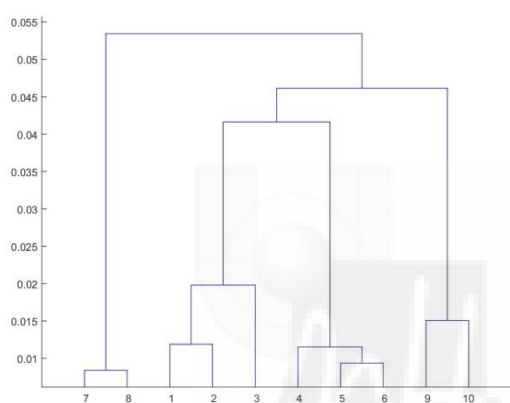
2 ventanas y 16 máscaras



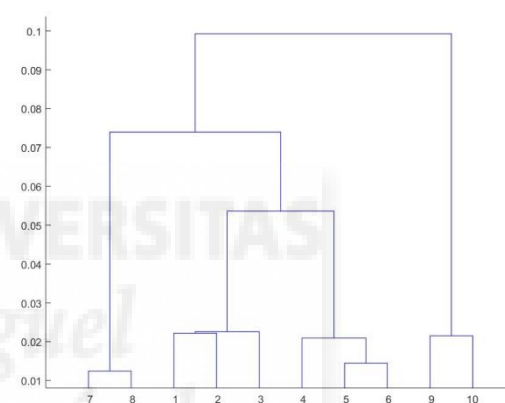
4 ventanas y 4 máscaras



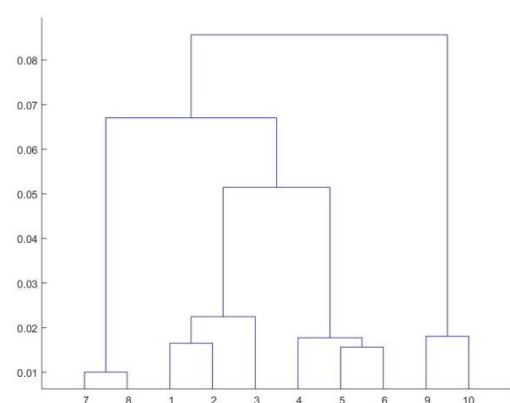
4 ventanas y 8 máscaras



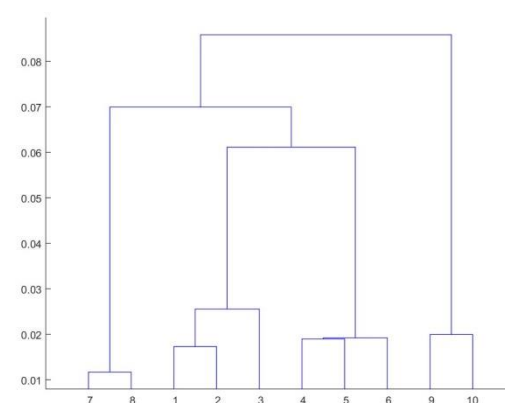
4 ventanas y 16 máscaras



8 ventanas y 4 máscaras



8 ventanas y 8 máscaras



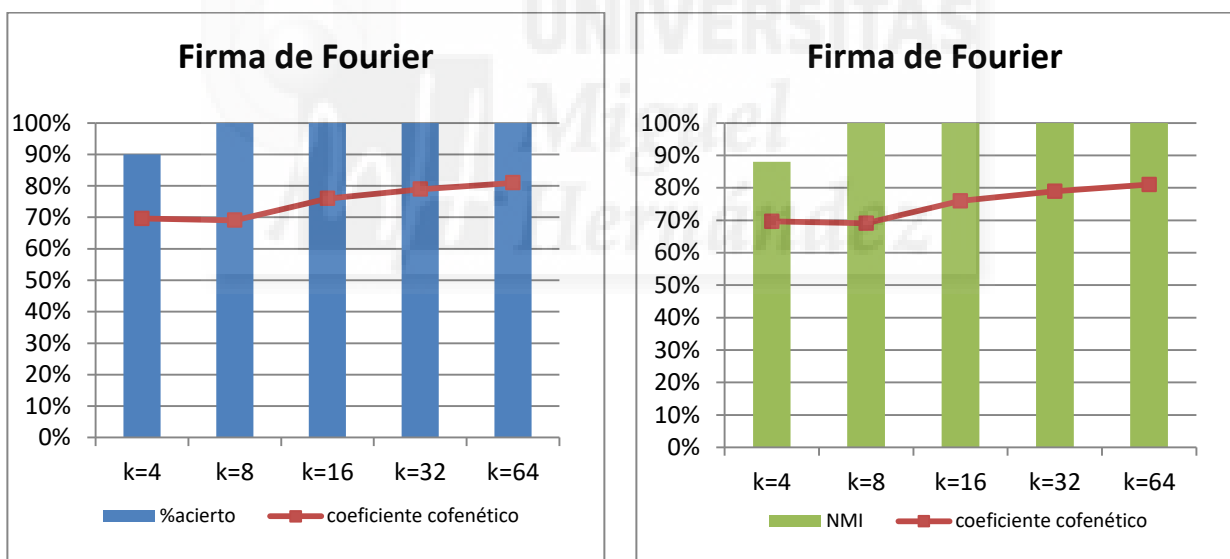
8 ventanas y 16 máscaras

Para el caso de los dendrogramas obtenidos al emplear Gist como descriptor, se puede observar cómo los resultados en los que se han empleado 4 ventanas son muy parecidos entre sí salvo en la altura de las barras verticales. Exactamente lo mismo ocurre para los dendrogramas obtenidos empleando 8 ventanas. En cuanto a los resultados mostrados para 1 ventana, se puede apreciar que la variación entre emplear 4 u 8 máscaras es mínima, mientras que para 16 máscaras el orden de agrupamiento varía. Lo mismo ocurre para los resultados obtenidos empleando 2 ventanas.

Una vez hemos obtenido los dendrogramas, éstos se seccionan mediante la función *cluster* y se asignan los objetos que quedan por debajo de cada corte a un cluster individual. Dado que se han seleccionado imágenes de las 5 estancias (4 habitaciones y pasillo) de la planta, el número de clusters indicado en la función *cluster* será 5.

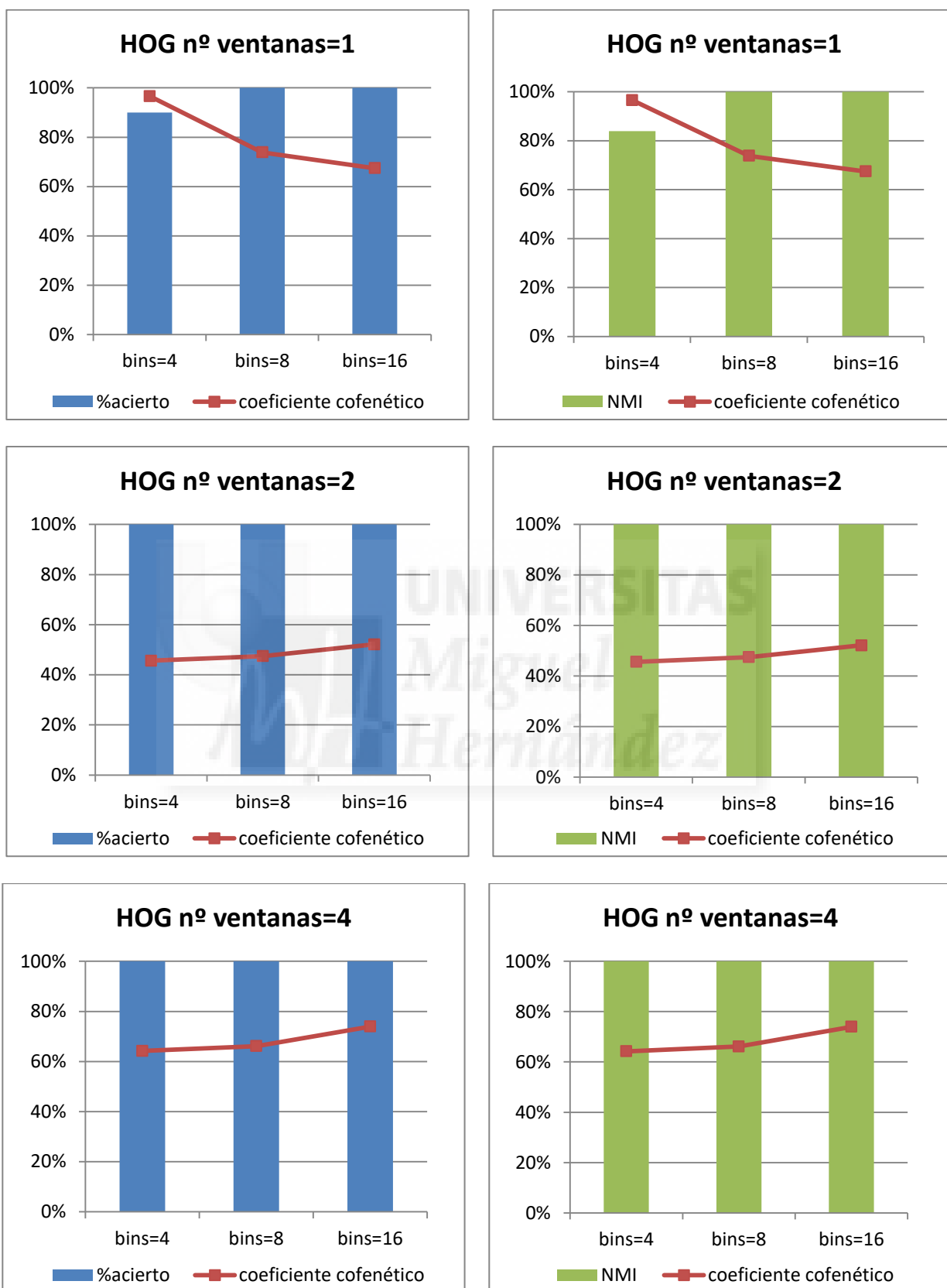
Finalmente, el último paso, es comprobar cómo de efectivos han sido los métodos empleados en cuanto a precisión se refiere. A continuación, se muestran las gráficas en la que se indica la precisión, es decir, el porcentaje de acierto calculado mediante la función *success* y la función *NMI*. Superpuesto en las mismas gráficas, se representa el coeficiente de correlación cofenético el cual compara la distancia real entre imágenes y la distancia cofenética e indica cómo de bien se han agrupado las escenas. Cuanto más cercano a la unidad es este coeficiente, más preciso es el clustering.

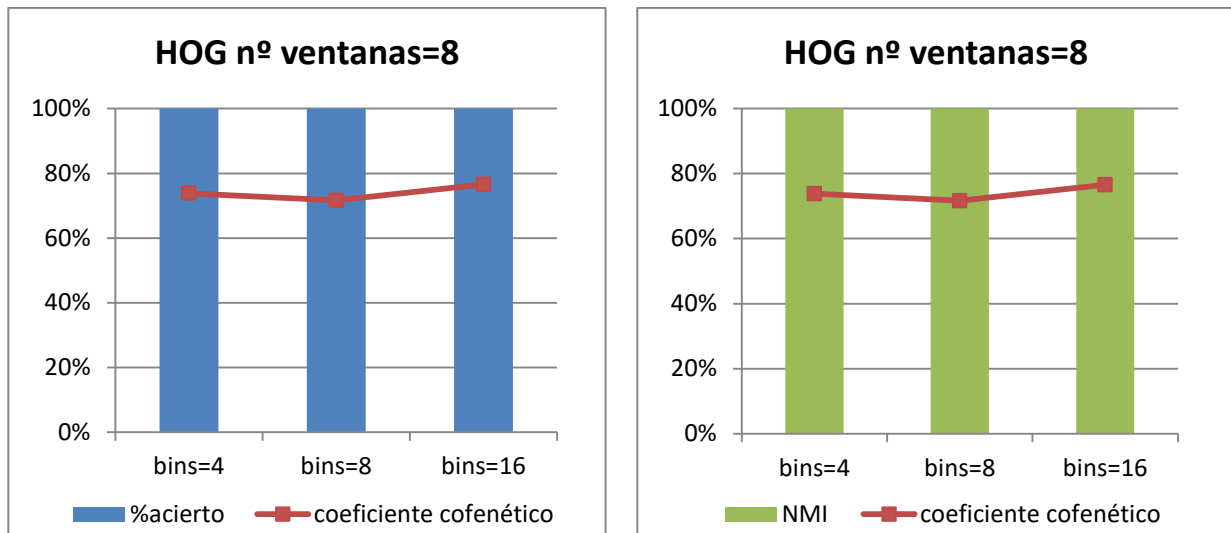
Resultados obtenidos empleando como descriptor la Firma de Fourier:



Las gráficas muestran que la Firma de Fourier ofrece muy buenos resultados para valores de k superiores a 8 ya que el acierto es del 100% y el coeficiente de correlación cofenético se encuentra en torno al 0,8 para valores de k superiores a 16.

Resultados obtenidos empleando como descriptor HOG:

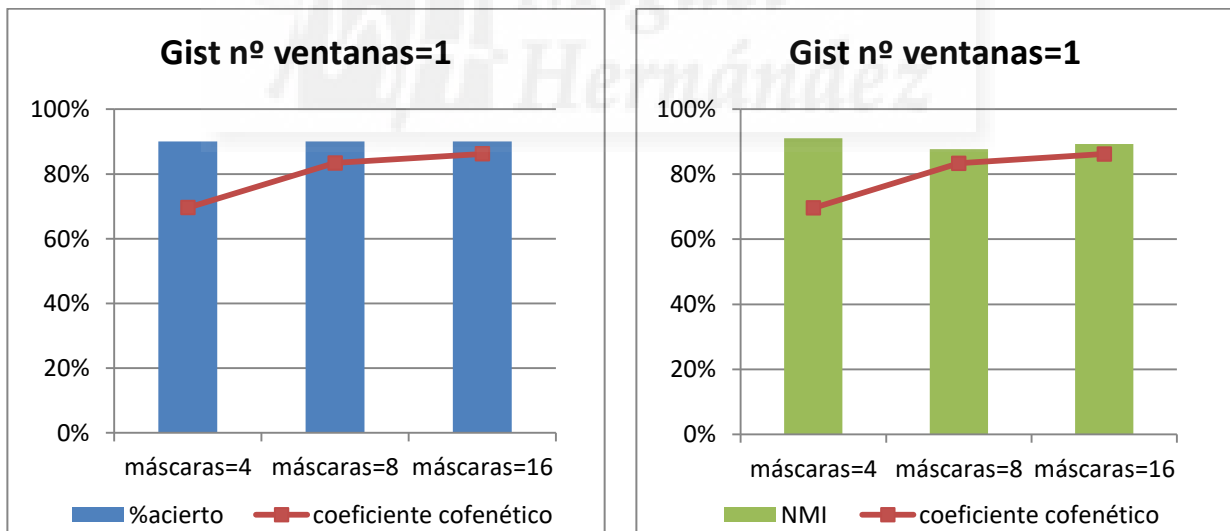


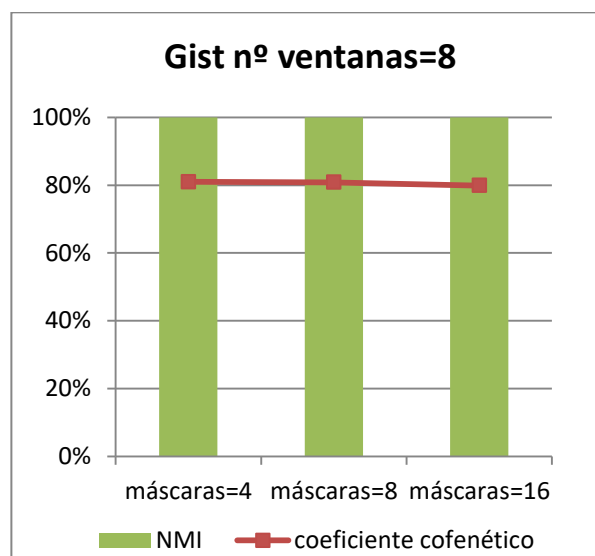
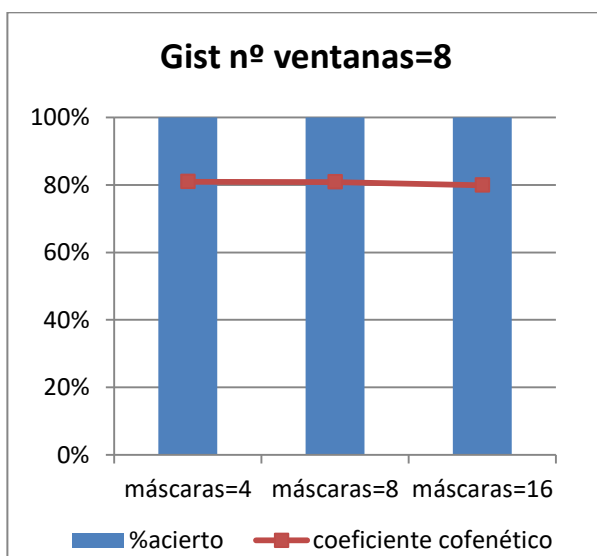
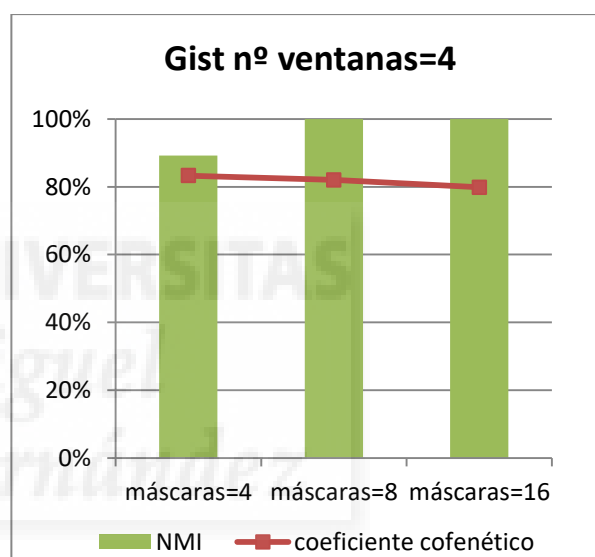
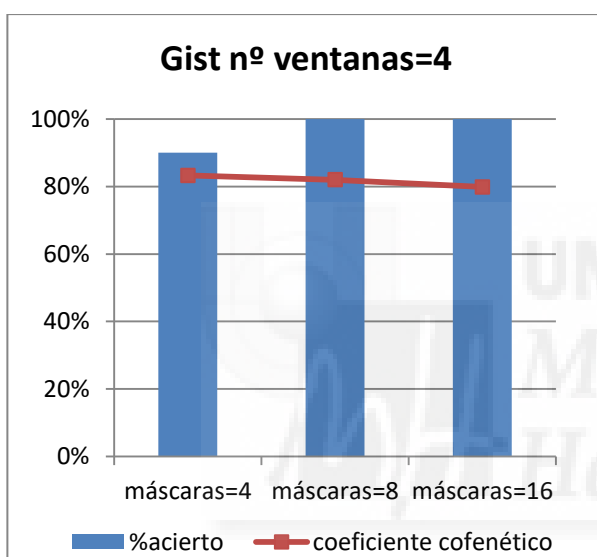
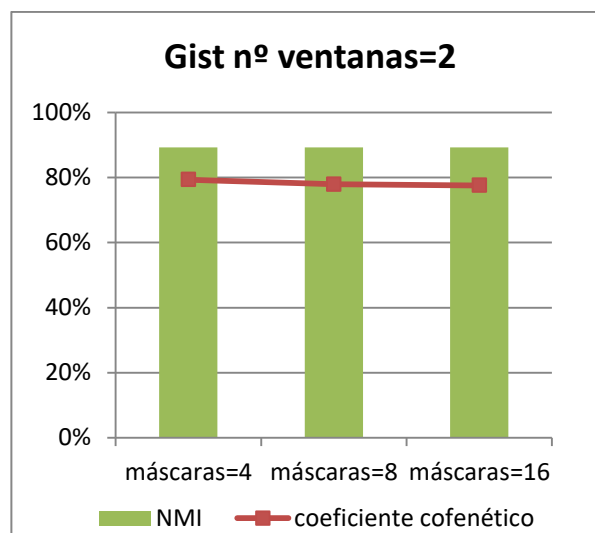
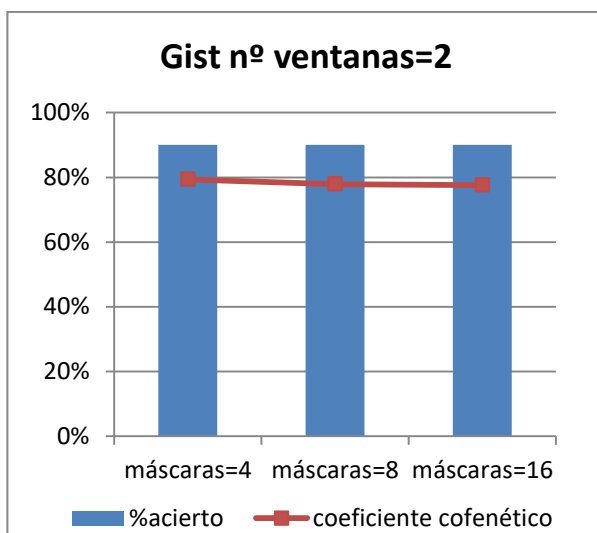


En las gráficas obtenidas empleando HOG, se puede ver que el porcentaje de acierto es del 100% para todos los casos excepto para el caso de 1 ventana y 4 bins.

En cambio, el coeficiente de correlación cofenético se queda por debajo del 0,5 para el caso de 2 ventanas, en torno a 0,6-0,7 para 4 ventanas y alrededor de 0,7 para las 8 ventanas. Para el caso de 1 ventana y 4 bins se obtiene el mejor valor de este coeficiente siendo superior a 0,95, en cambio el valor va descendiendo para el resto de casos de 1 ventana conforme aumenta el número de bins.

Resultados obtenidos empleando como descriptor Gist:





En las gráficas obtenidas empleando Gist como descriptor, se puede apreciar que el porcentaje de acierto ronda el 90% para los casos de 1 y 2 ventanas y a partir de 4 ventanas y 8 máscaras se alcanza el 100%.

En cuanto al coeficiente de correlación cofenético, se observa que a medida que aumenta el número de ventanas y el número de máscaras tiene tendencia a disminuir ligeramente su valor.

5.6. Resultados obtenidos para el conjunto completo de escenas

En este apartado se van a analizar los resultados obtenidos para el conjunto completo de imágenes. Como se comentaba en el apartado 5.1, para reducir el tiempo de cálculo y trabajar de una manera más cómoda, se ha decidido emplear 1 de cada 4 imágenes de la base de datos, siendo un total de 266 imágenes las que se han manejado para el estudio de los algoritmos.

Las imágenes corresponden a una planta del Centro de Investigación Alemán de Inteligencia Artificial en Saarbrücken. Las estancias que encontramos en esta planta son una cocina, un baño, un despacho, un cuarto de impresoras y el pasillo.

Tal y como se mencionaba en el apartado 3.4 se van a seguir dos métodos para la creación del mapa jerárquico topológico. Para el primero de los métodos se va a emplear en primer lugar el algoritmo de clustering jerárquico y posteriormente el algoritmo de clustering espectral. En el segundo de los métodos se va a emplear únicamente el algoritmo de clustering espectral.

5.6.1. Primera fase: Clustering jerárquico

En esta primera fase se pretende conseguir un mapa de alto nivel en el que los clusters obtenidos se correspondan con las diferentes estancias de la planta.

En primer lugar se calculan los descriptores de apariencia visual global para cada una de las escenas. Como se comentaba en el capítulo 2, los descriptores empleados en este trabajo son la Firma de Fourier, el Histograma de Orientación del Gradiente y Gist.

Para el descriptor basado en la Firma de Fourier, se varía el parámetro k que es el número de columnas de la matriz de módulos que retenemos. Los valores para k con los que trabajamos son 4, 8, 16, 32 y 64.

Para el descriptor basado en el Histograma de Orientación del Gradiente, los parámetros a configurar son el número de celdas o ventanas horizontales y el número de bins o canales en cada histograma. Los números de ventanas horizontales con los que trabajamos son 1, 2, 4 y 8 y para cada una de ellos analizamos los resultados con 4, 8 y 16 bins.

Para el descriptor basado en Gist, los parámetros que se configuran son el número de máscaras o filtros de Gabor a aplicar y el número de ventanas horizontales que se emplean para reducir la información. Los valores con los que trabajamos son 4, 8 y 16 filtros de Gabor y 1, 2, 4 y 8 ventanas horizontales.

Una vez creados los descriptores, se comienza a realizar la clasificación en clusters con el objetivo de agrupar las escenas por estancias. Para encontrar la similitud o la disimilitud entre cada pareja de escenas empleamos la función *pdist*. Se estudian los métodos descritos en el apartado 4.1.1.

- Distancia Euclídea
- Distancia Cityblock
- Distancia Minkowski
- Distancia Chebychev
- Distancia Coseno
- Distancia Correlación
- Distancia Spearman

A continuación, se emplea la función *linkage* para analizar qué escenas deben ser agrupadas para formar los clusters. Se hace uso de los métodos de medición de distancia entre clusters detallados en el apartado 4.1.2.

- Average. Método de la distancia media no ponderada.
- Weighted. Método de la distancia media ponderada.
- Centroid. Método de la distancia al centro de masas sin ponderar.
- Median. Método de la distancia al centro de masas ponderada.
- Single. Método de la distancia más cercana.
- Complete. Método de la distancia más lejana.
- Ward. Método de la mínima varianza.

El siguiente paso es generar los dendrogramas correspondientes a cada configuración de parámetros. Más adelante se mostrarán los dendrogramas correspondientes a los métodos que mayor porcentaje de acierto ofrecen.

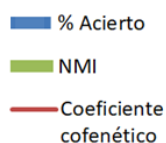
Una vez obtenidos los dendrogramas, éstos se seccionan mediante la función *cluster* y se asignan los objetos que quedan por debajo de cada corte a un cluster. En un primer estudio, se van a indicar en la función *cluster* 5 clusters dado que, en la planta, hay 5 estancias (4 habitaciones y el pasillo).

El último paso es comprobar la precisión de los métodos empleados. A continuación, se muestran las gráficas en las que se indica la precisión, es decir, el porcentaje de acierto calculado mediante la función *success* y la función *NMI*. Superpuesto en las mismas gráficas, se

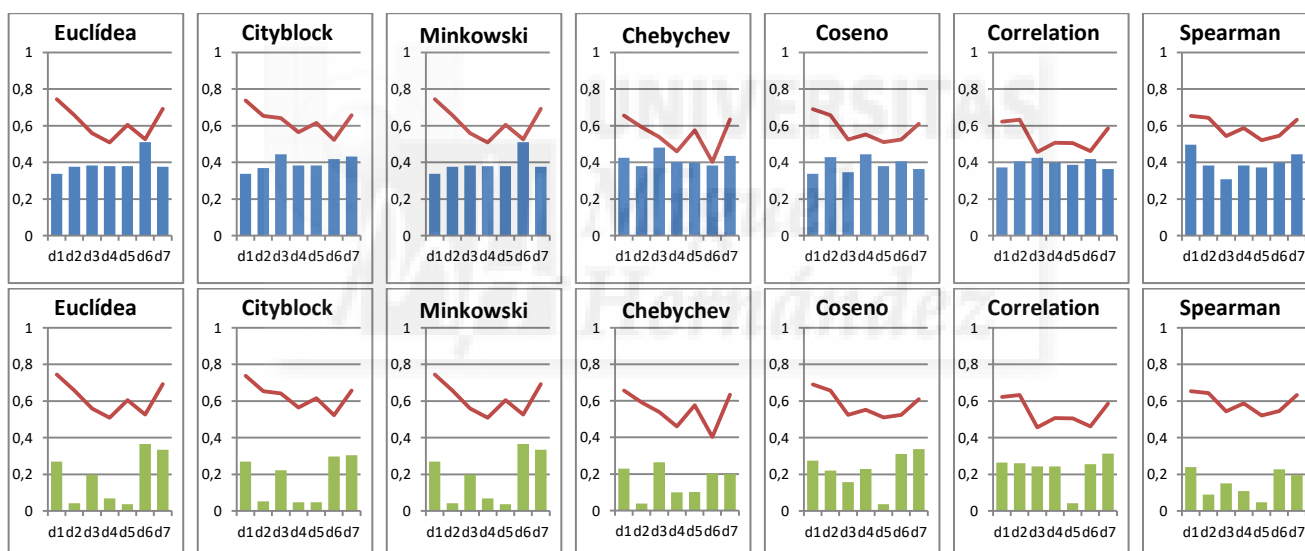
representa el coeficiente de correlación cofenético. Para las gráficas, se han codificado los métodos de medición de distancia de la siguiente manera:

- d1: Average
- d2: Centroid
- d3: Complete
- d4: Median
- d5: Single
- d6: Ward
- d7: Weighted

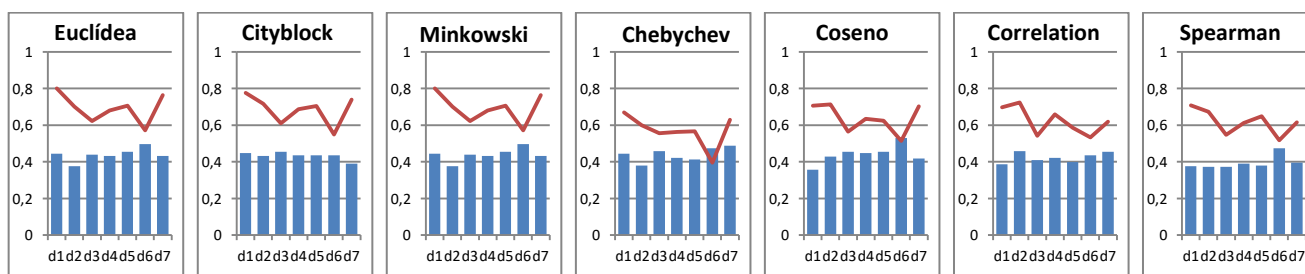
Legenda empleada:

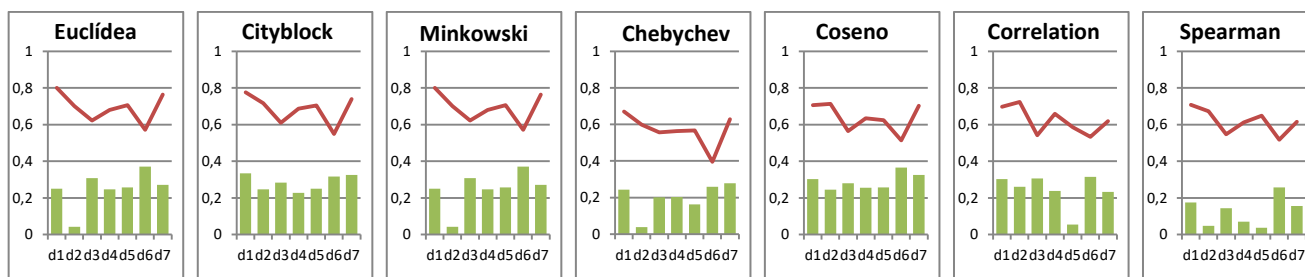


Resultados obtenidos empleando la Firma de Fourier para k=4

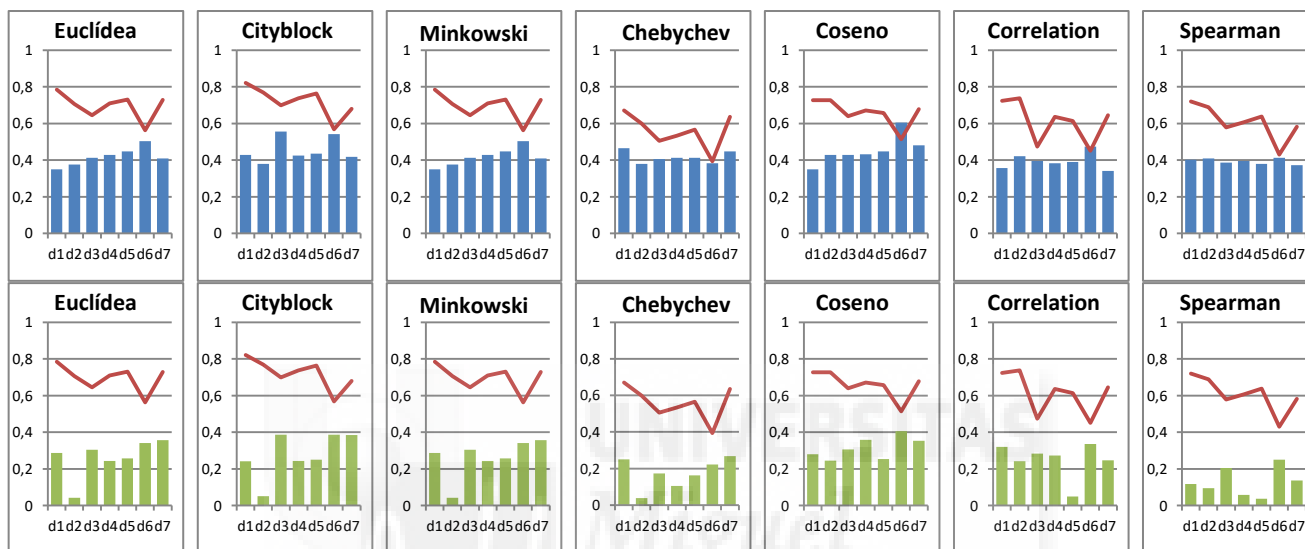


Resultados obtenidos empleando la Firma de Fourier para k=8

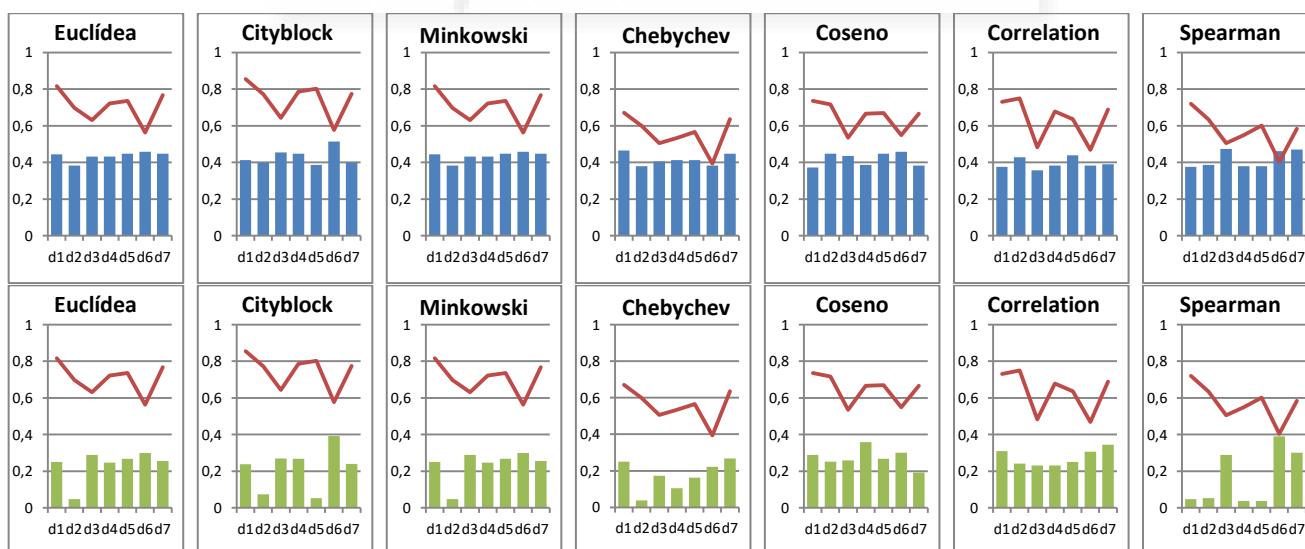




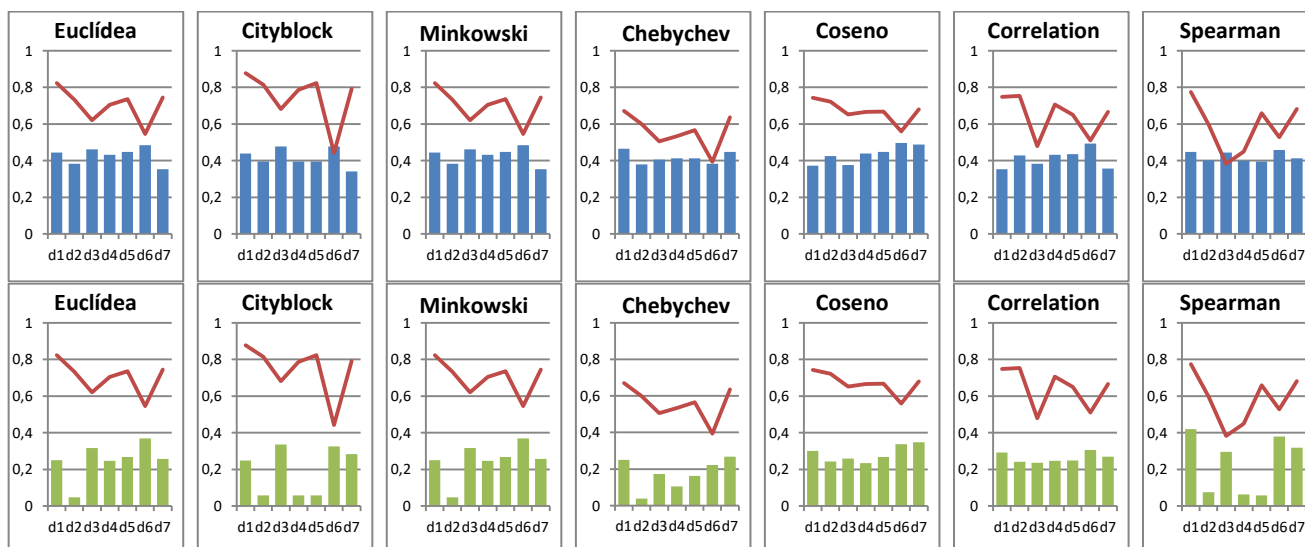
Resultados obtenidos empleando la Firma de Fourier para k=16



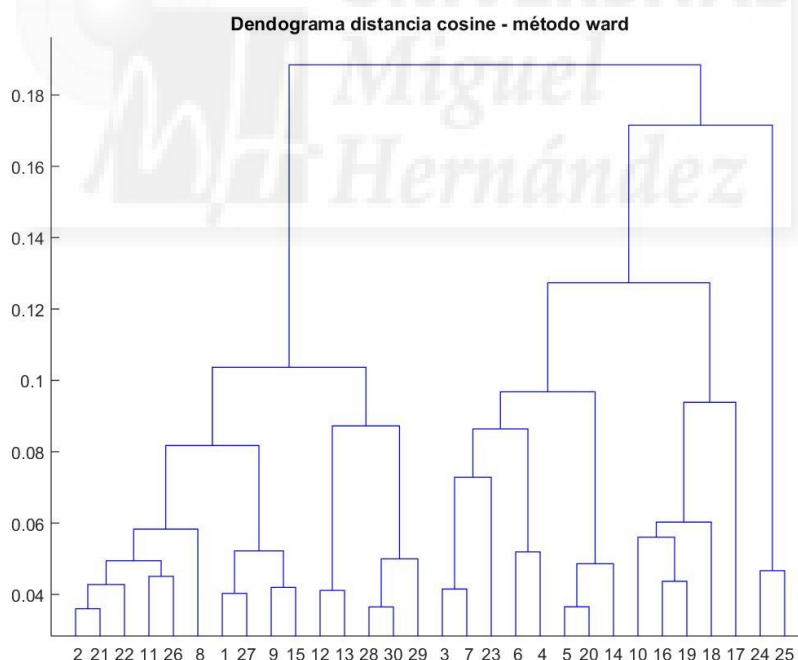
Resultados obtenidos empleando la Firma de Fourier para k=32



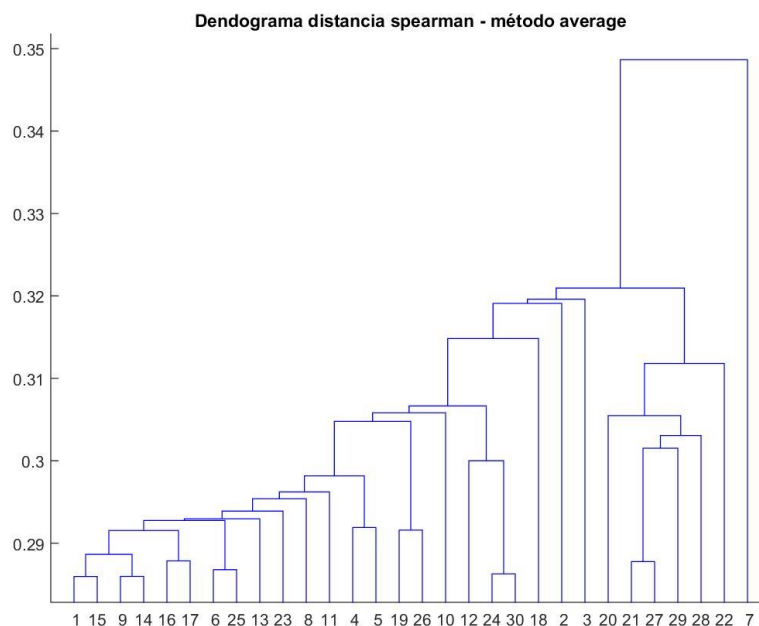
Resultados obtenidos empleando la Firma de Fourier para k=64



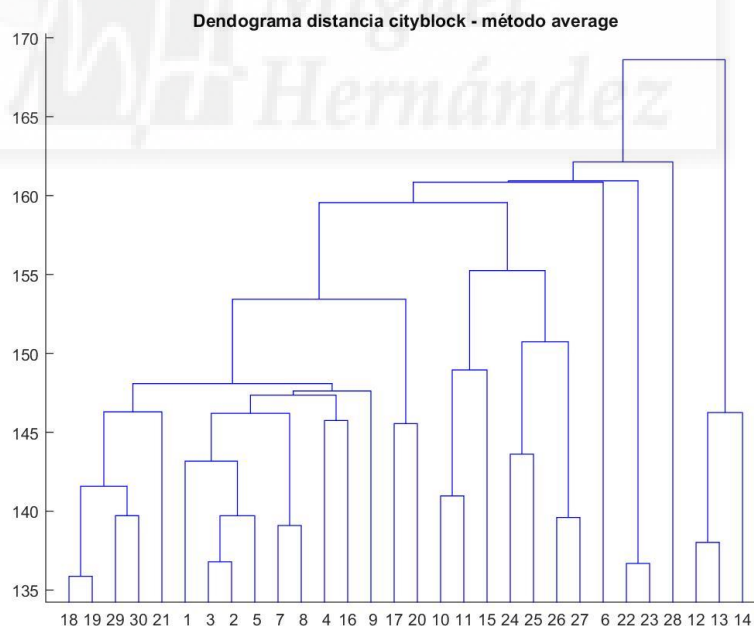
Como se puede apreciar en las gráficas, el porcentaje de acierto para la Firma de Fourier es bastante bajo ya que para todos los valores de k estudiados, el acierto ronda el 40%. El mejor resultado de todos se alcanza para k=16 con la distancia *Coseno* y el método *Ward* con un acierto del 60.52%. El dendrograma obtenido para este caso se muestra a continuación.



En cuanto los valores obtenidos de la función *NMI*, los resultados son muy bajos, estando alrededor del 20%, algunos de ellos no alcanzando ni el 5%. El valor más elevado para *NMI* obtenido con la Firma de Fourier es 41.96% para la distancia *Spearman*, el método *Average* y k=64. El dendrograma obtenido para este caso se muestra a continuación.

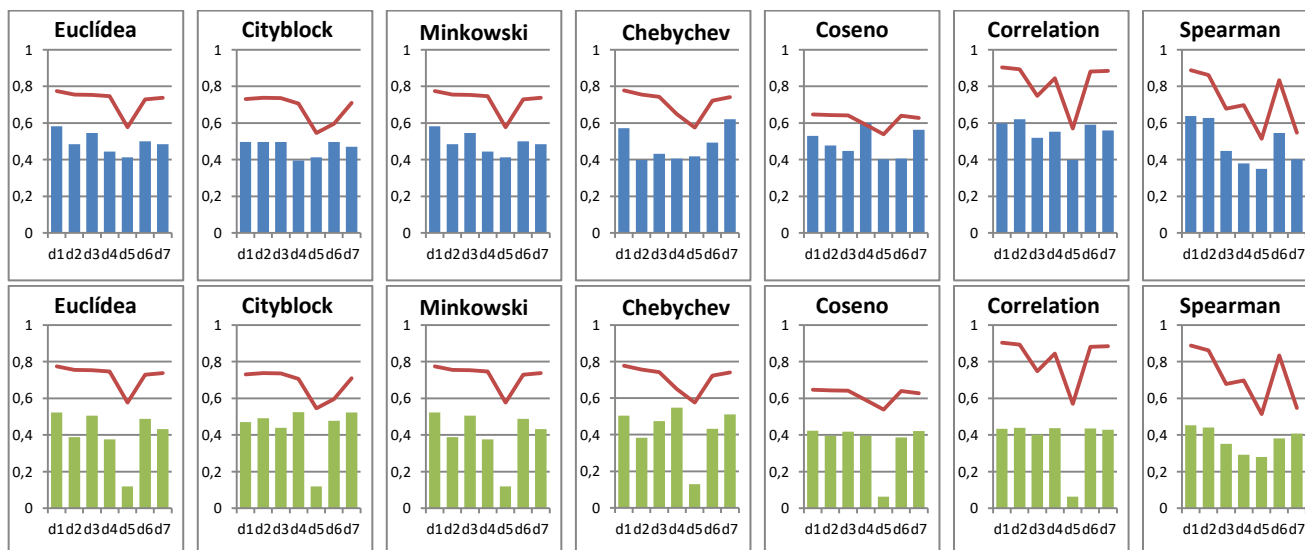


En lo referente al coeficiente cofenético, se obtienen los valores más altos el método *Average*, seguido de los métodos *Centroid*, *Median*, *Single* y *Weighted*. El mayor valor de todos se alcanza para la distancia *Cityblock*, el método *Average* y $k=64$, siendo de 0.878. El dendrograma obtenido para este caso se muestra a continuación.

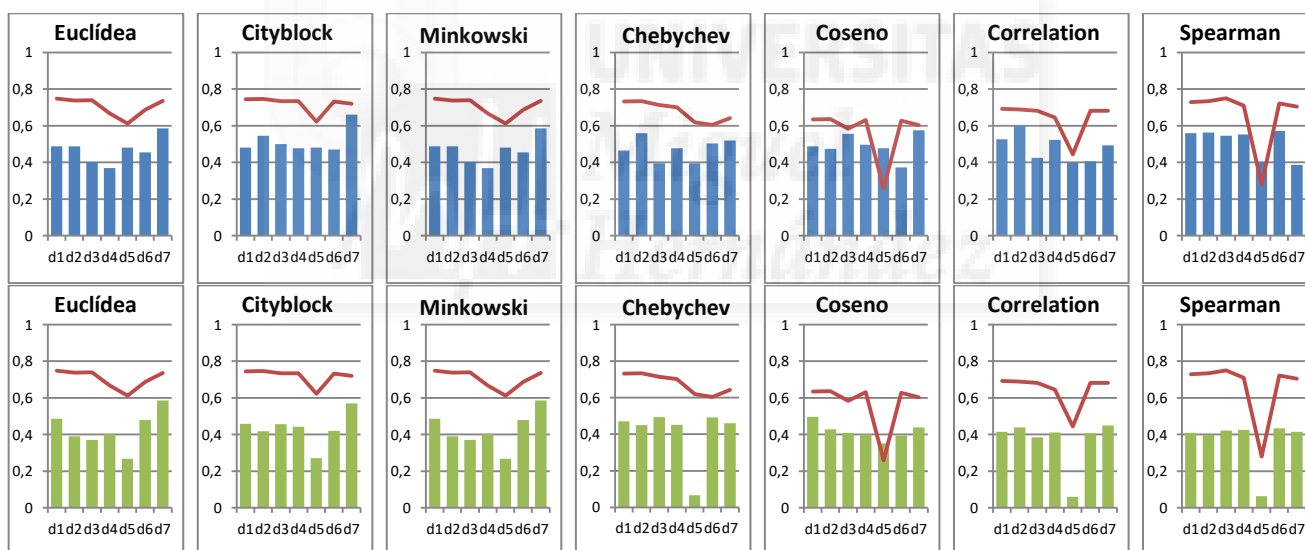


El tipo de distancia no influye en los resultados de la función *success* pero sí en los resultados obtenidos con *NMI*, siendo peores los resultados para el tipo *Spearman*. Lo mismo sucede para el método de medición de distancia, obteniéndose los peores resultados para el método *Centroid*. El aumento del valor k en el descriptor tampoco se aprecia en los resultados obtenidos.

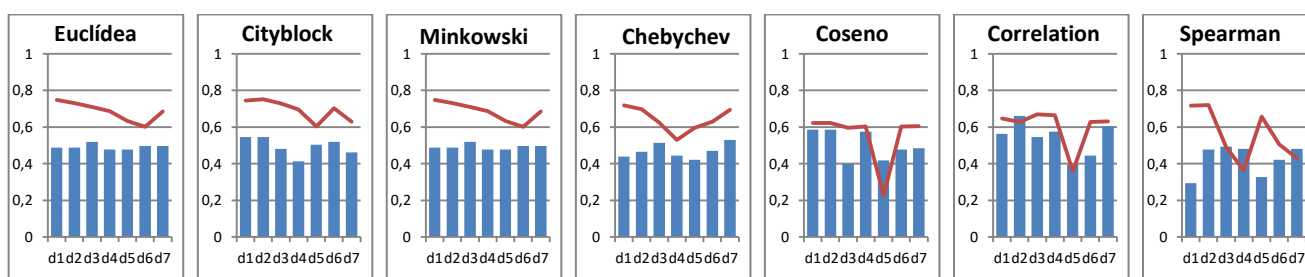
Resultados obtenidos empleando HOG para 1 ventana y 4 bins

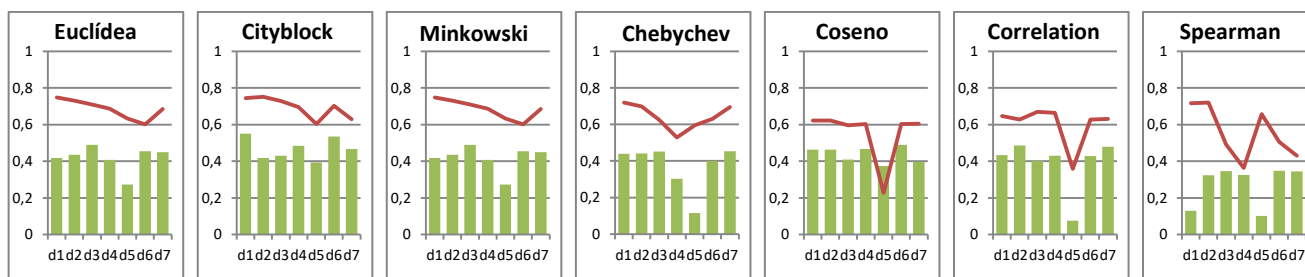


Resultados obtenidos empleando HOG para 1 ventana y 8 bins

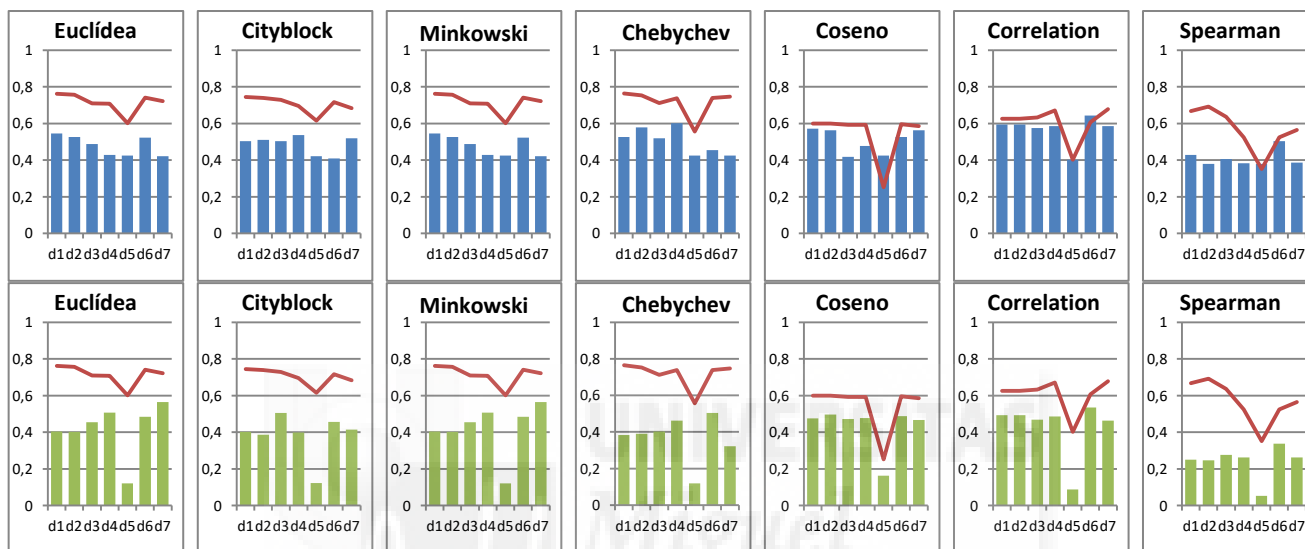


Resultados obtenidos empleando HOG para 1 ventana y 16 bins

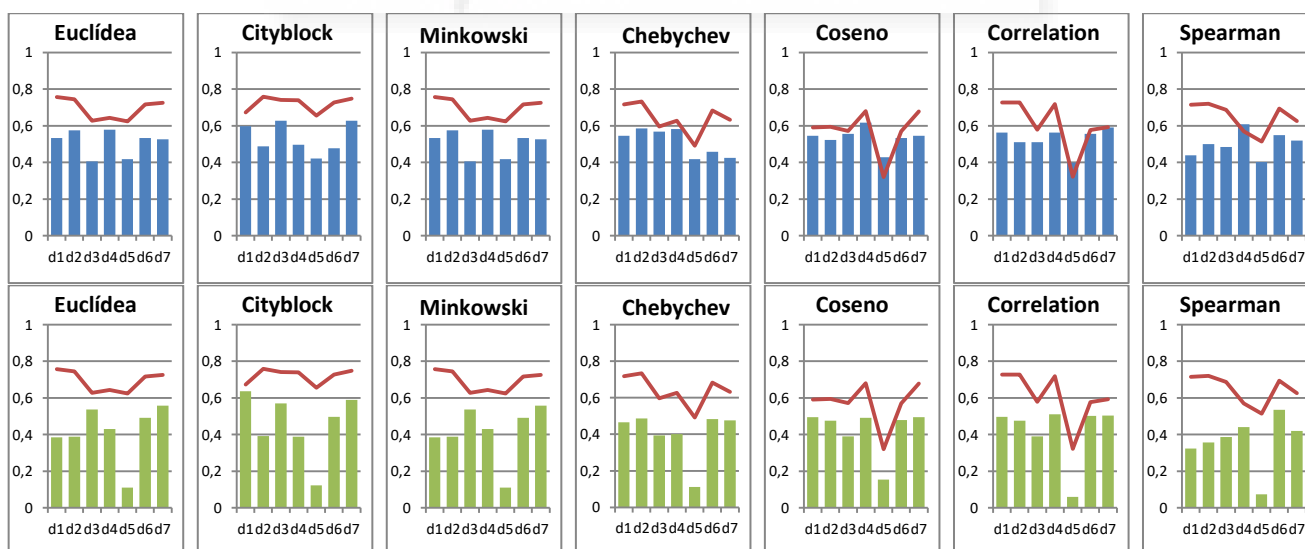




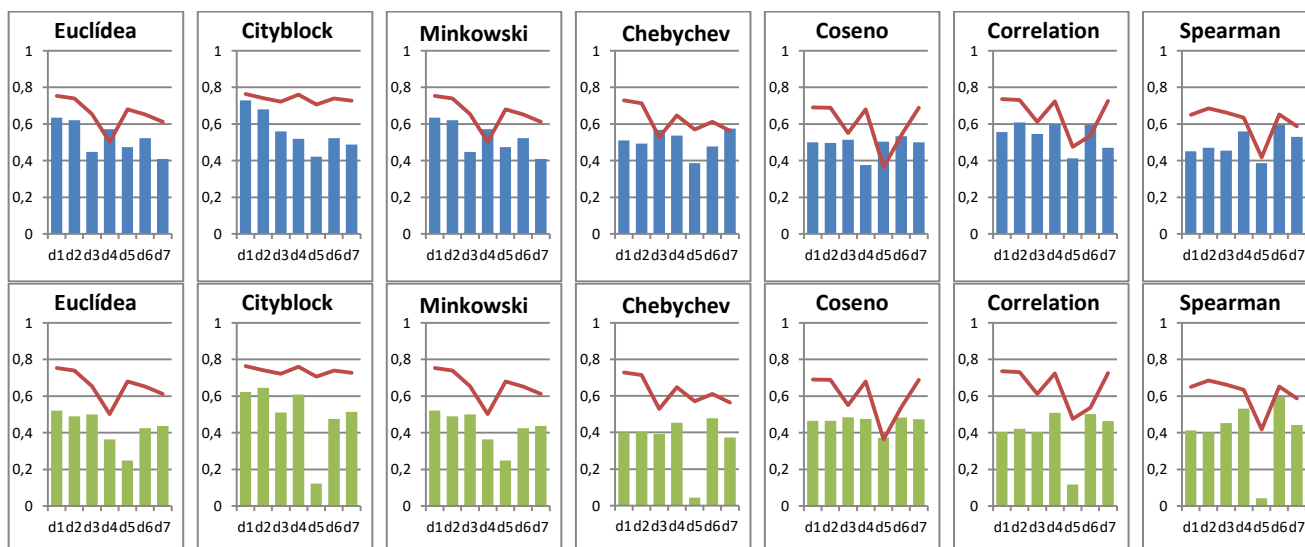
Resultados obtenidos empleando HOG para 2 ventanas y 4 bins



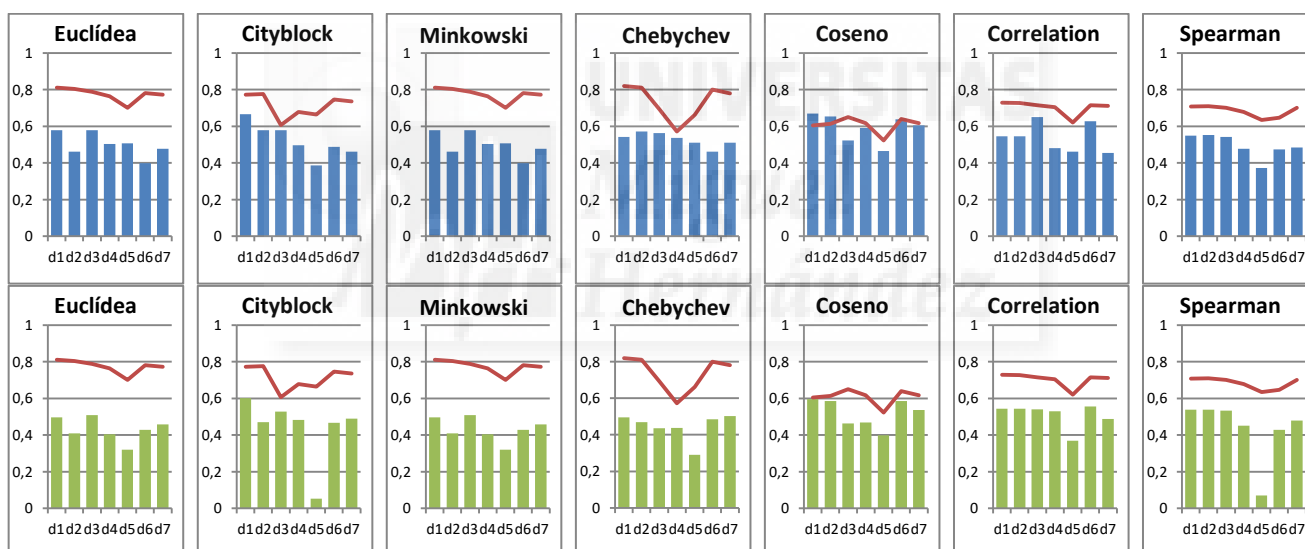
Resultados obtenidos empleando HOG para 2 ventanas y 8 bins



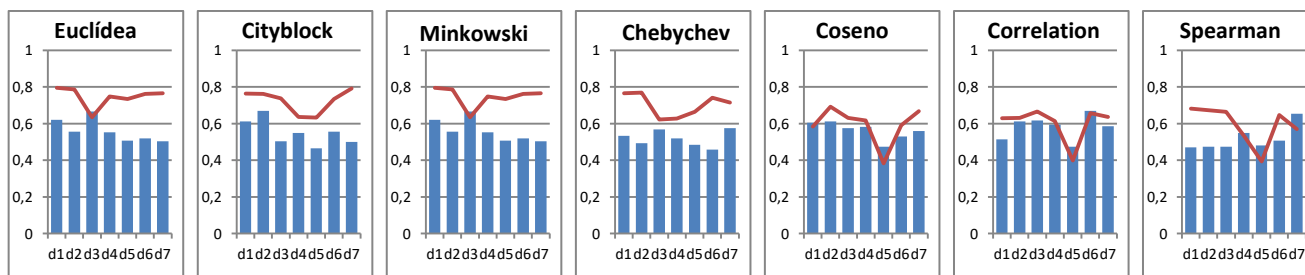
Resultados obtenidos empleando HOG para 2 ventanas y 16 bins

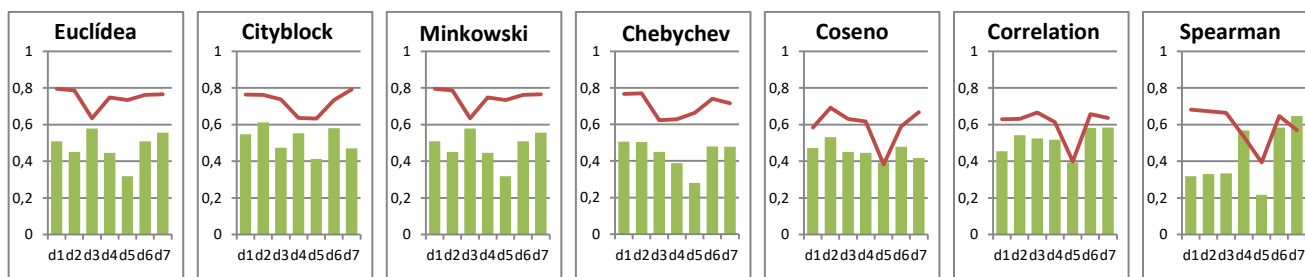


Resultados obtenidos empleando HOG para 4 ventanas y 4 bins

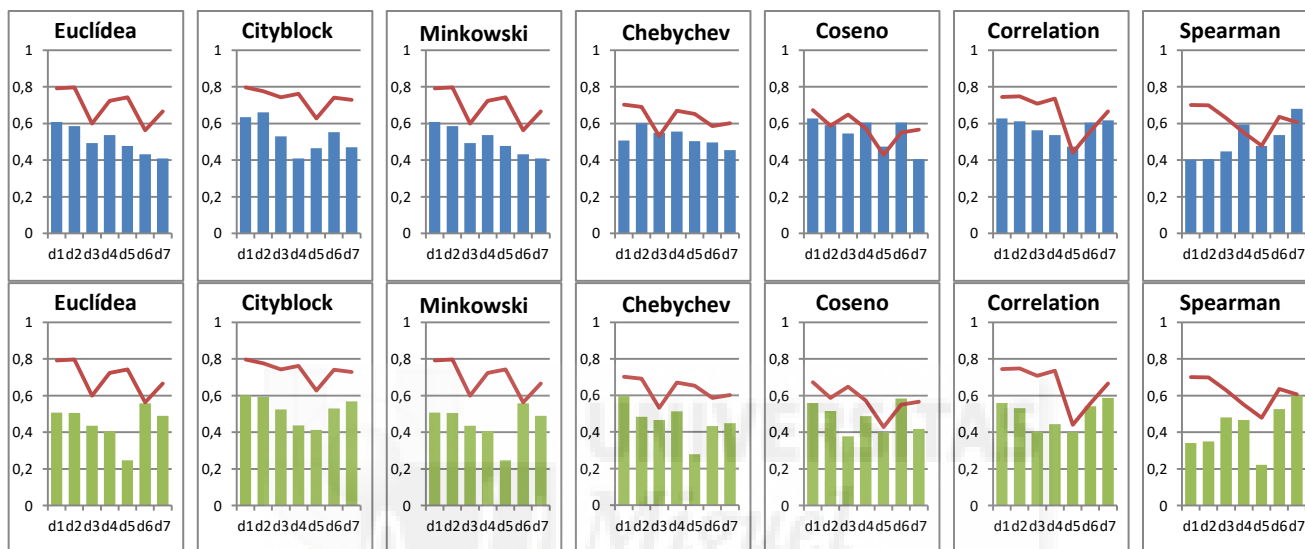


Resultados obtenidos empleando HOG para 4 ventanas y 8 bins

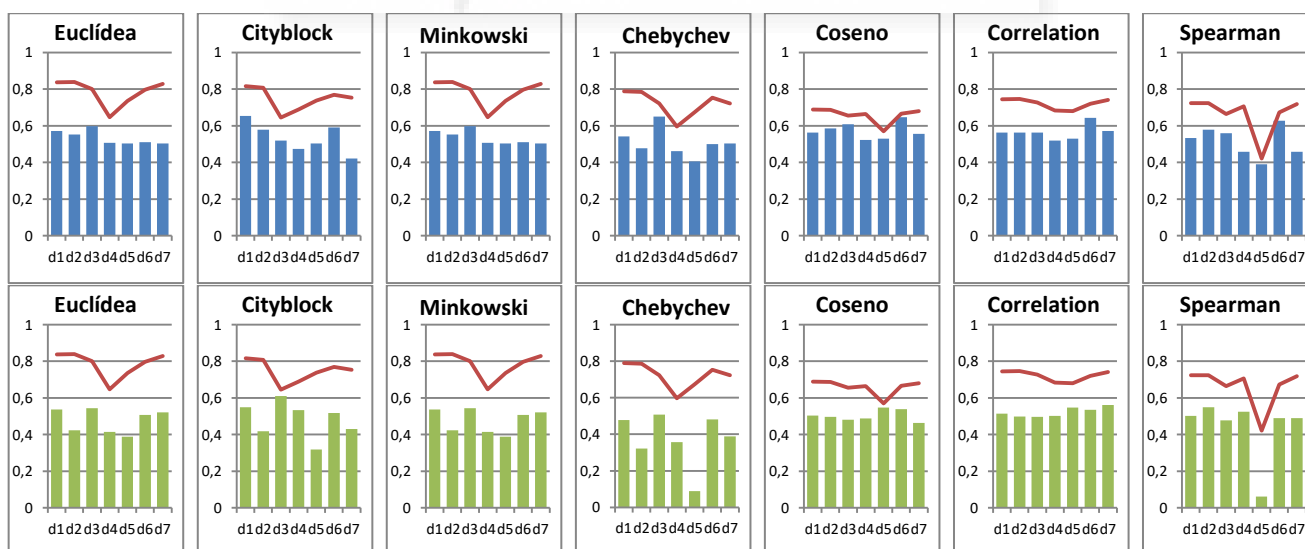




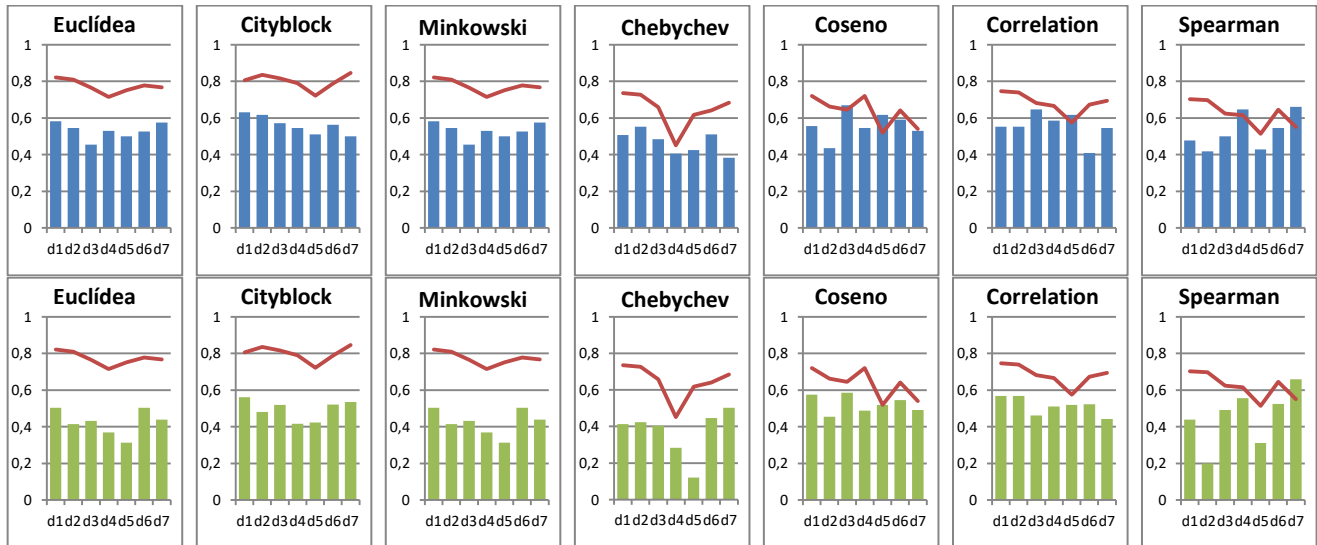
Resultados obtenidos empleando HOG para 4 ventanas y 16 bins



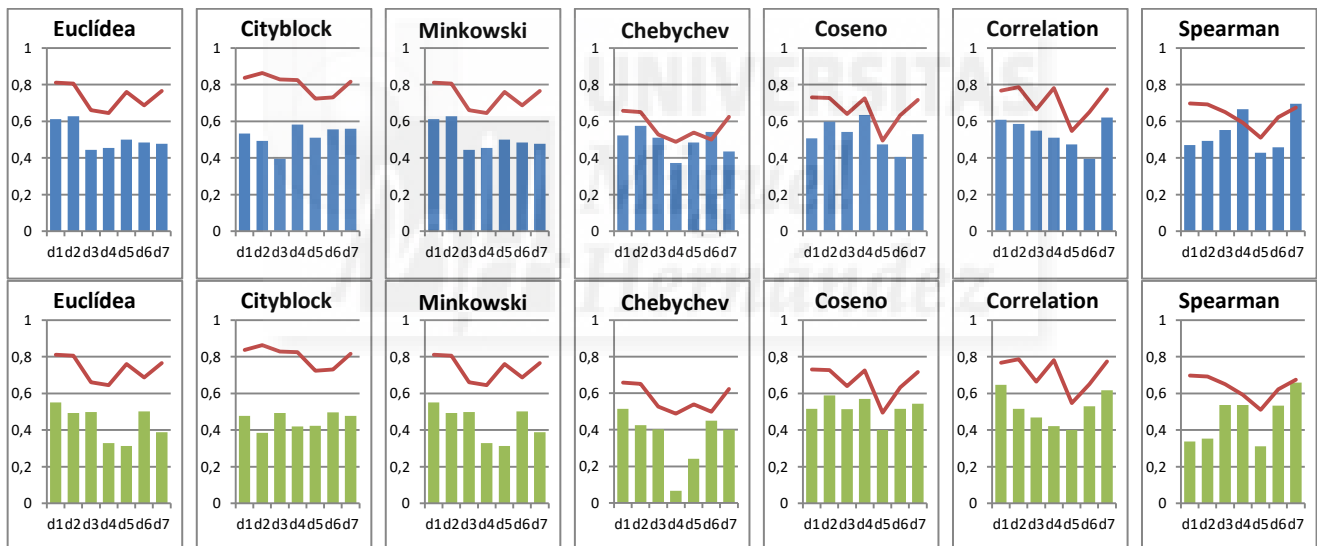
Resultados obtenidos empleando HOG para 8 ventanas y 4 bins



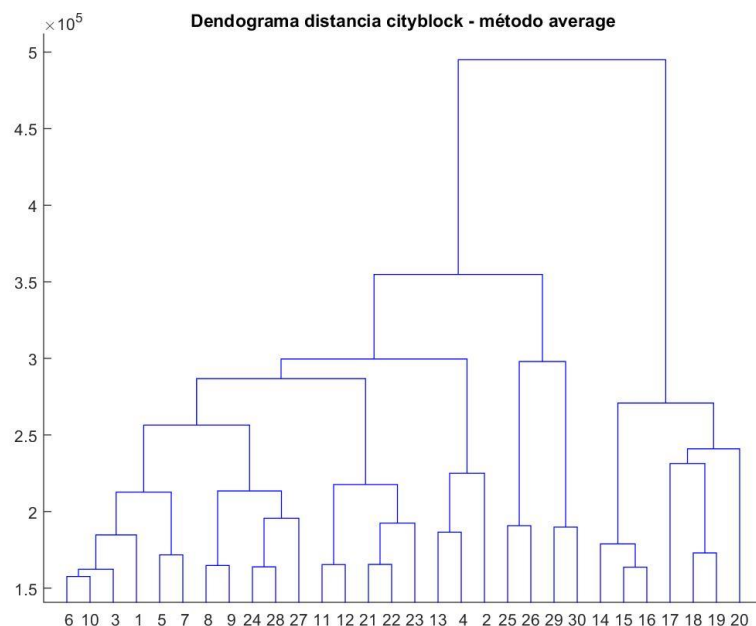
Resultados obtenidos empleando HOG para 8 ventanas y 8 bins



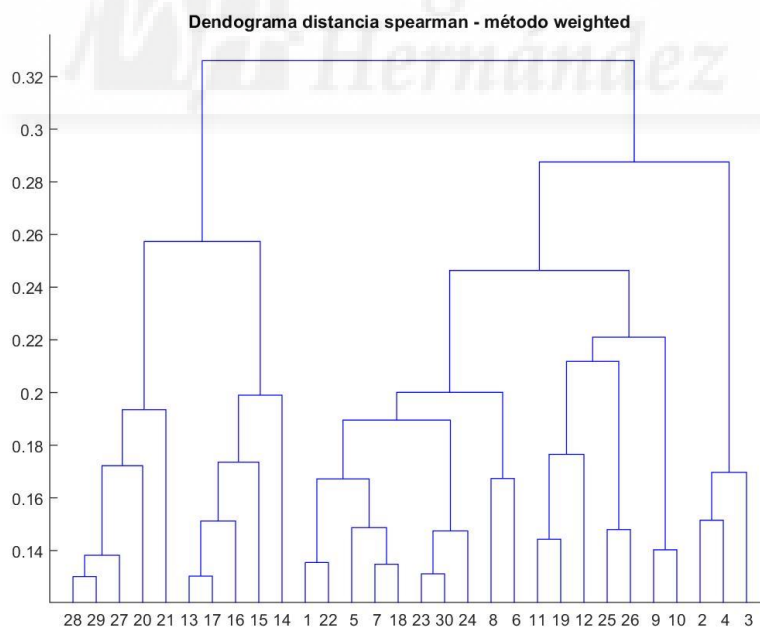
Resultados obtenidos empleando HOG para 8 ventanas y 16 bins



En el caso de emplear HOG como descriptor, apreciamos un aumento de la precisión obteniendo un acierto medio en torno a 50%-55%. A medida que aumentan el número de ventanas y el número de bins el porcentaje de acierto incrementa ligeramente. Aun así no conseguimos un resultado aceptable para realizar la clasificación. El mejor de los resultados se alcanza para 2 ventanas y 16 bins empleando la distancia *Cityblock* y el método *Average* con un acierto del 72.93%. El dendrograma obtenido para este caso se muestra a continuación.

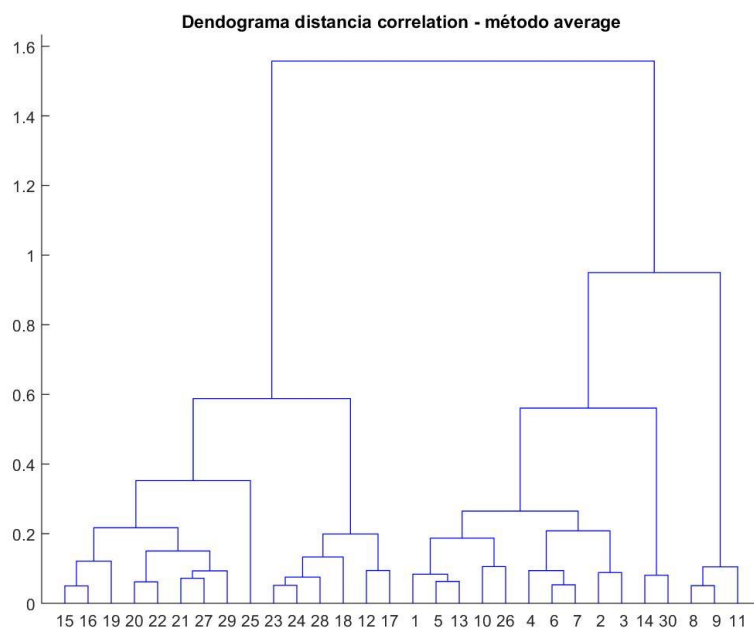


En cuanto los valores obtenidos de la función *NMI*, los resultados son bastante mejores que los obtenidos con la Firma de Fourier, estando la mayoría de ellos, alrededor del 40%-50%. El valor más elevado para *NMI* se consigue para la distancia *Spearman*, el método *Weighted* y 8 ventanas y 16 bins, siendo de 65.95%. El dendrograma obtenido para este caso se muestra a continuación.



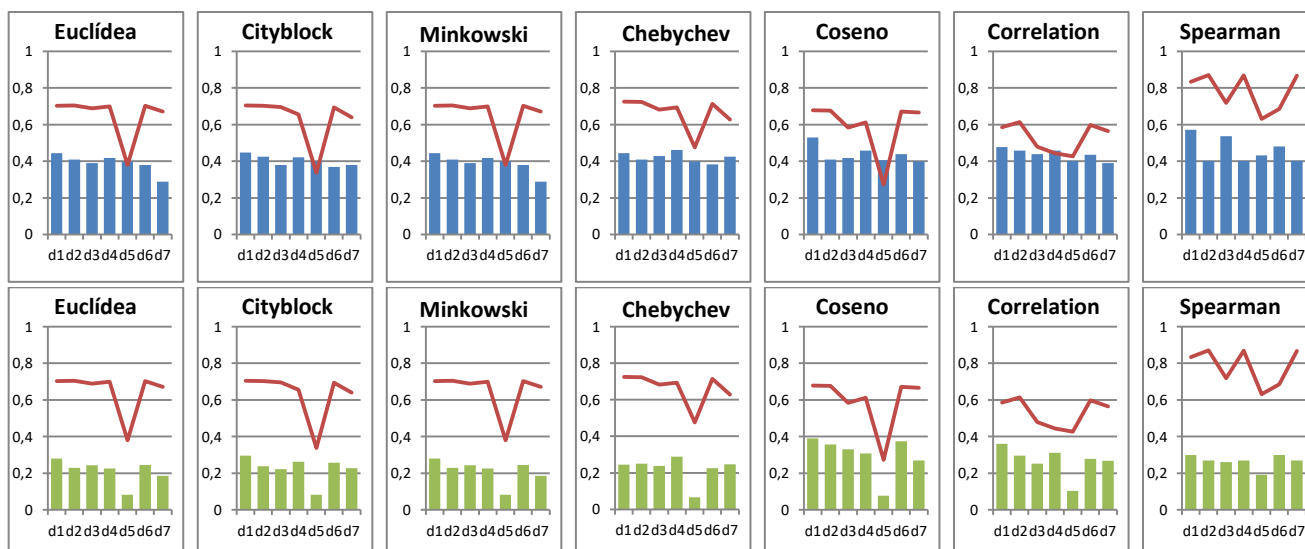
El coeficiente de correlación cofenético se encuentra en torno a un 0.65-0.7. Para un mismo número de ventanas, si aumentamos el número de bins el valor de este coeficiente disminuye ligeramente. El mejor resultado, 0.904, se obtiene con 1 ventana y 4 bins

empleando la distancia *Correlation* y el método *Average*. El dendrograma obtenido para este caso se muestra a continuación.

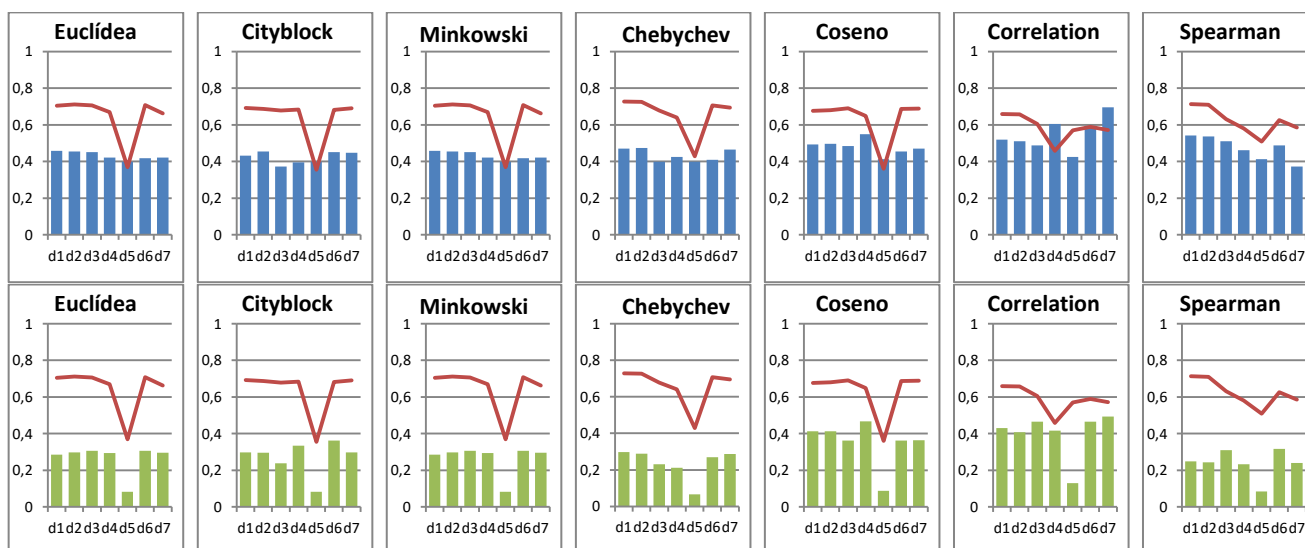


En general, los mejores resultados de la función *success* se obtienen con los tipos de distancia *Coseno* y *Correlation*. Los peores resultados para el método de medición de distancia, se obtienen para la función *NMI* con el método *Single*.

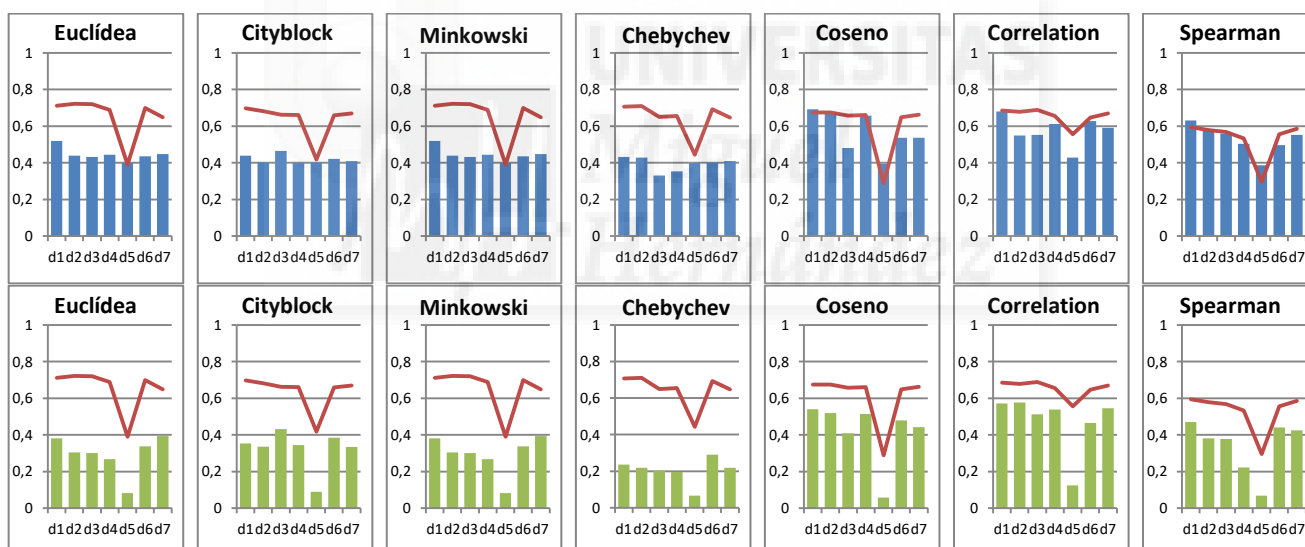
Resultados obtenidos empleando Gist para 1 ventana y 4 máscaras



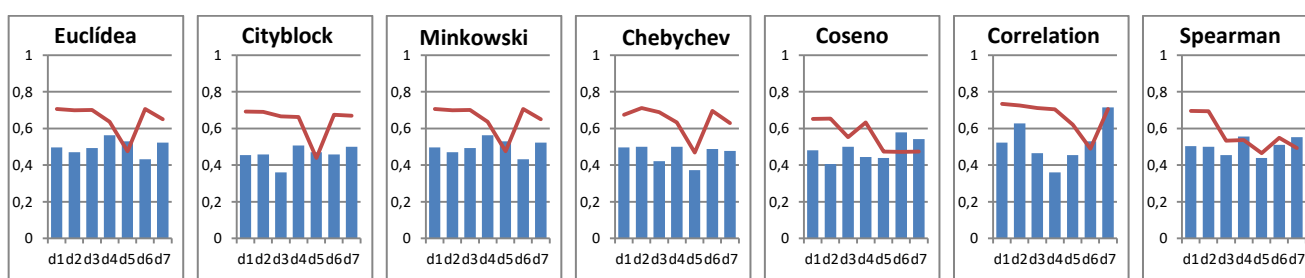
Resultados obtenidos empleando Gist para 1 ventana y 8 máscaras

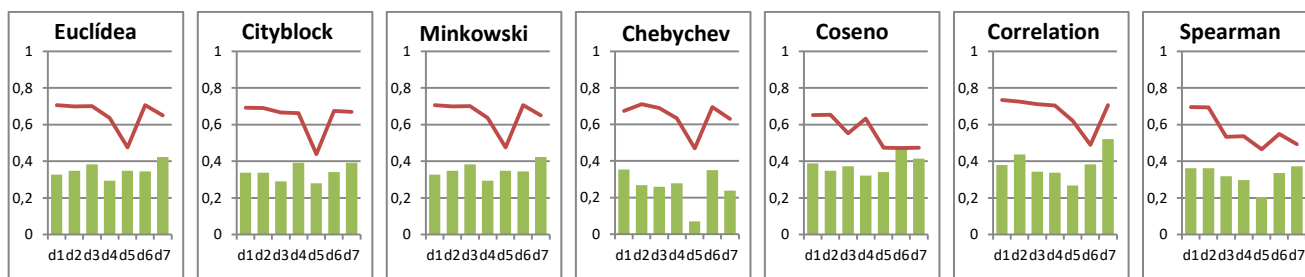


Resultados obtenidos empleando Gist para 1 ventana y 16 máscaras

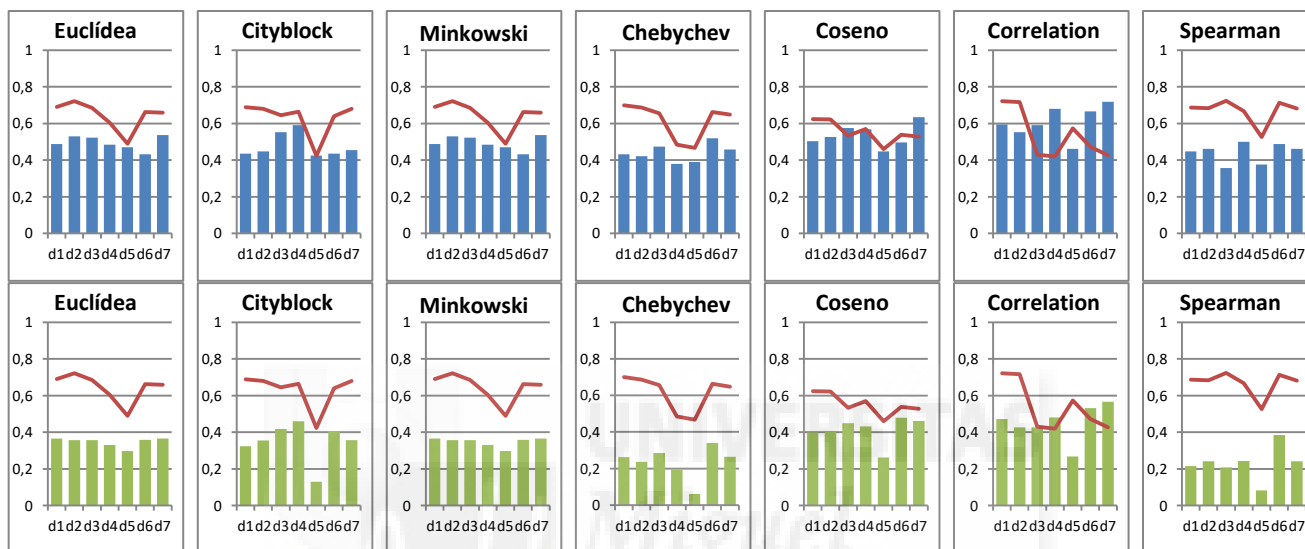


Resultados obtenidos empleando Gist para 2 ventanas y 4 máscaras

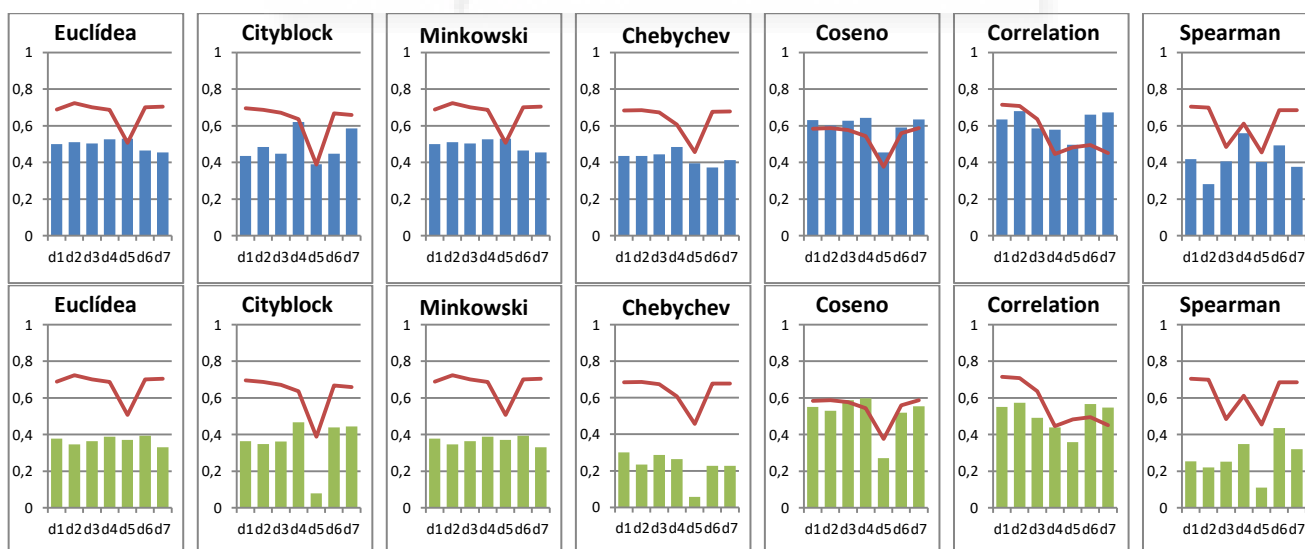




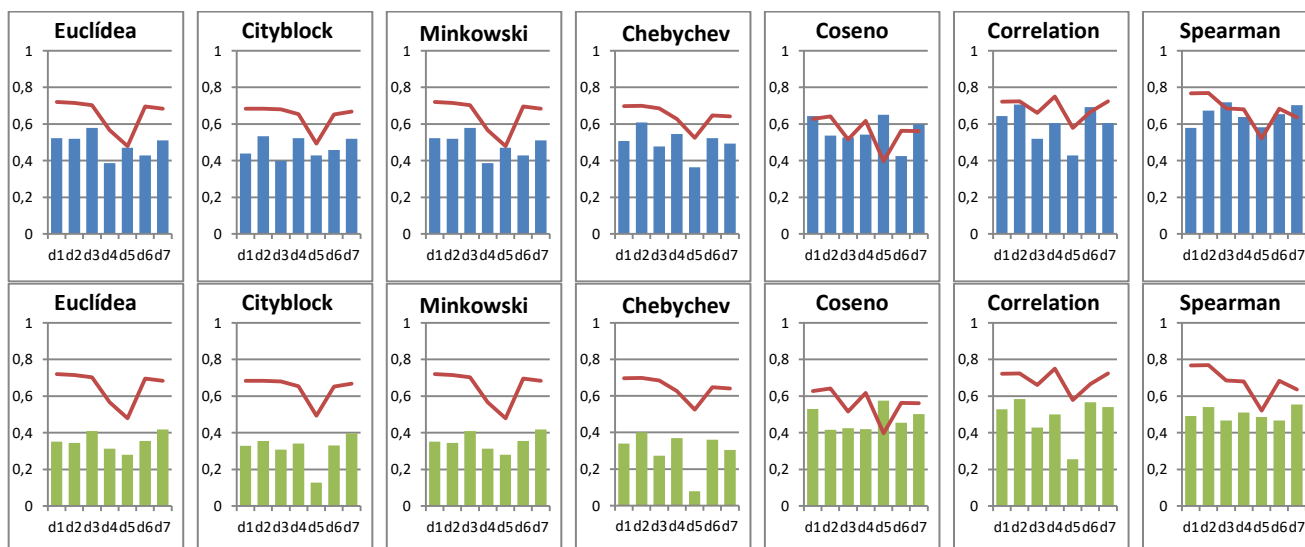
Resultados obtenidos empleando Gist para 2 ventanas y 8 máscaras



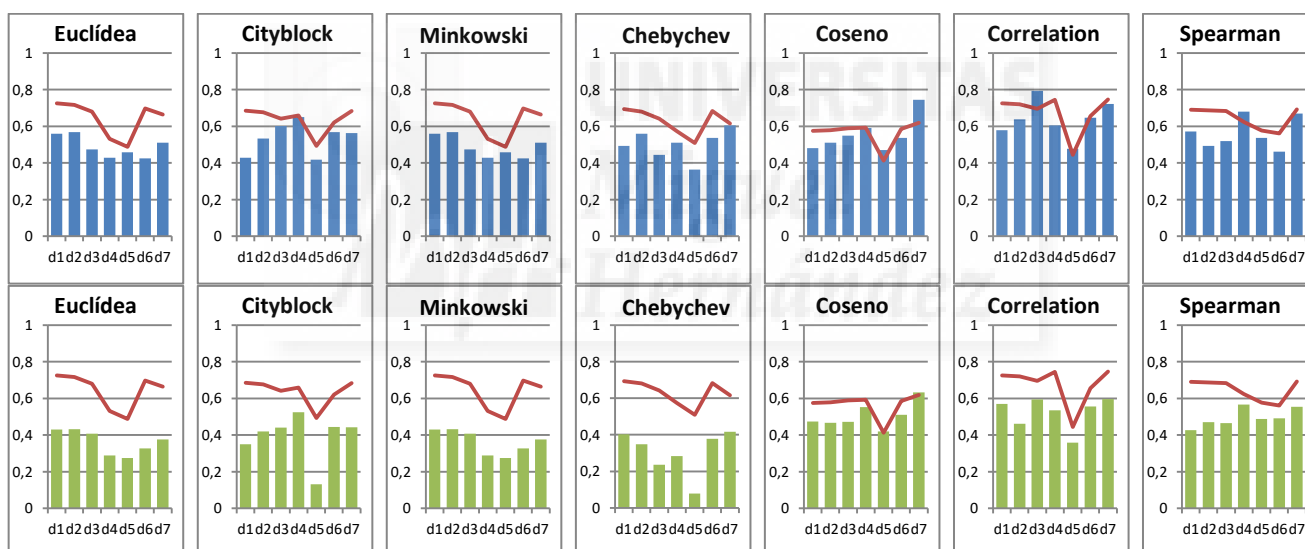
Resultados obtenidos empleando Gist para 2 ventanas y 16 máscaras



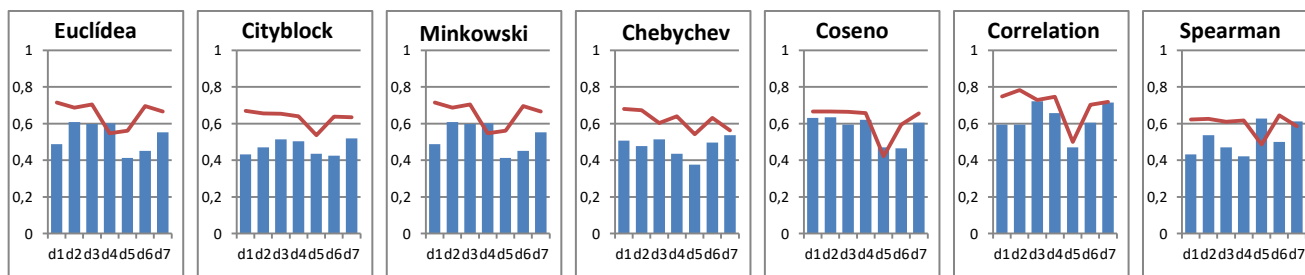
Resultados obtenidos empleando Gist para 4 ventanas y 4 máscaras

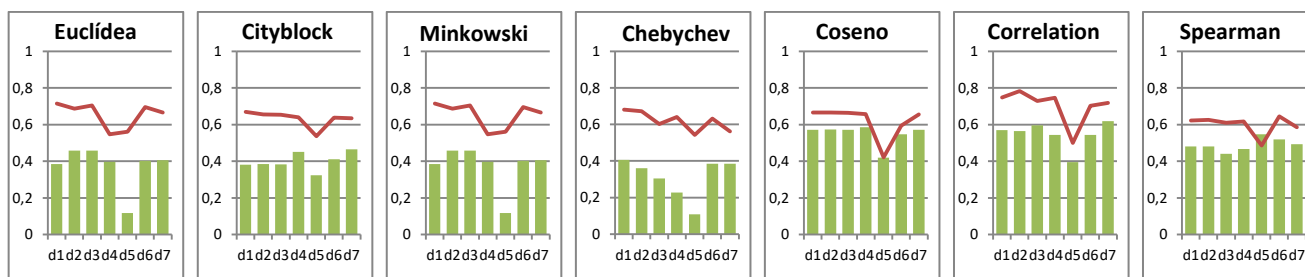


Resultados obtenidos empleando Gist para 4 ventanas y 8 máscaras

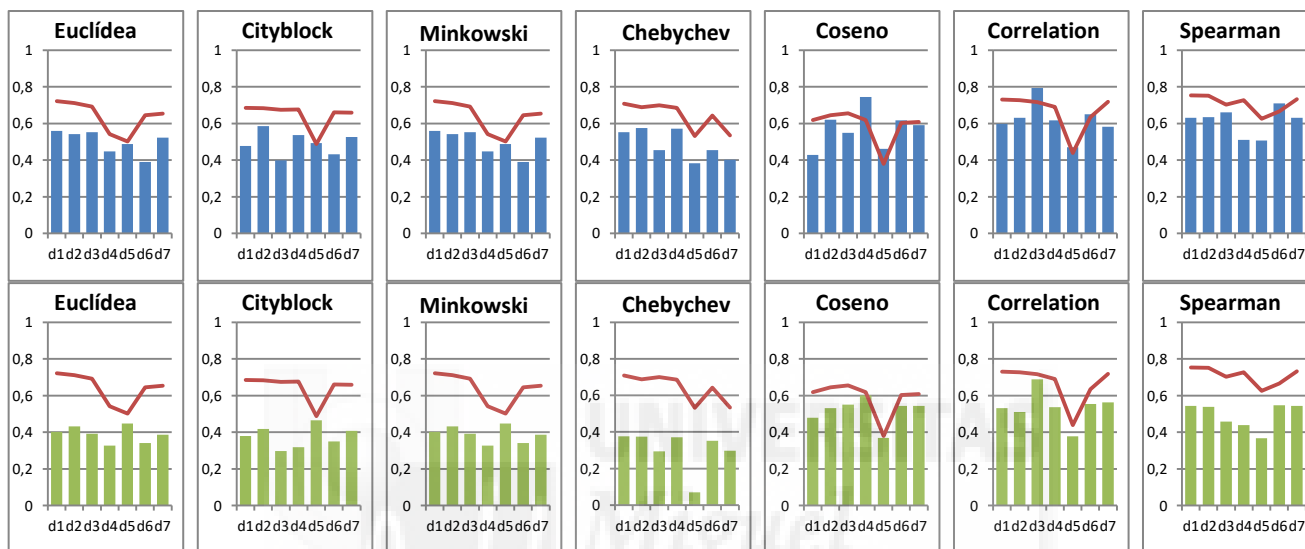


Resultados obtenidos empleando Gist para 4 ventanas y 16 máscaras

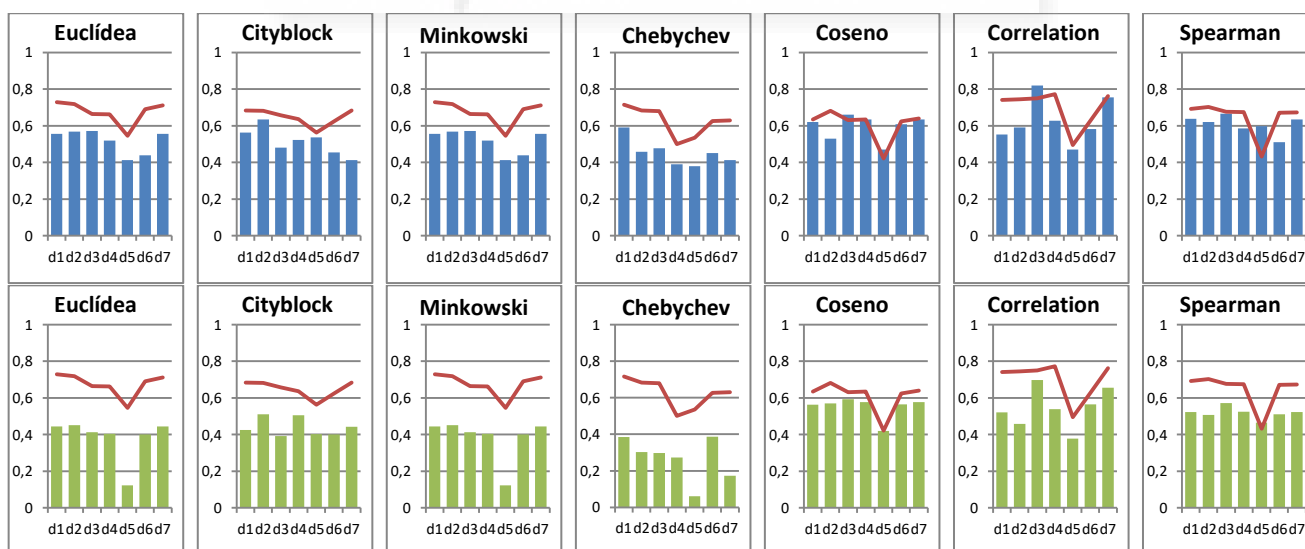




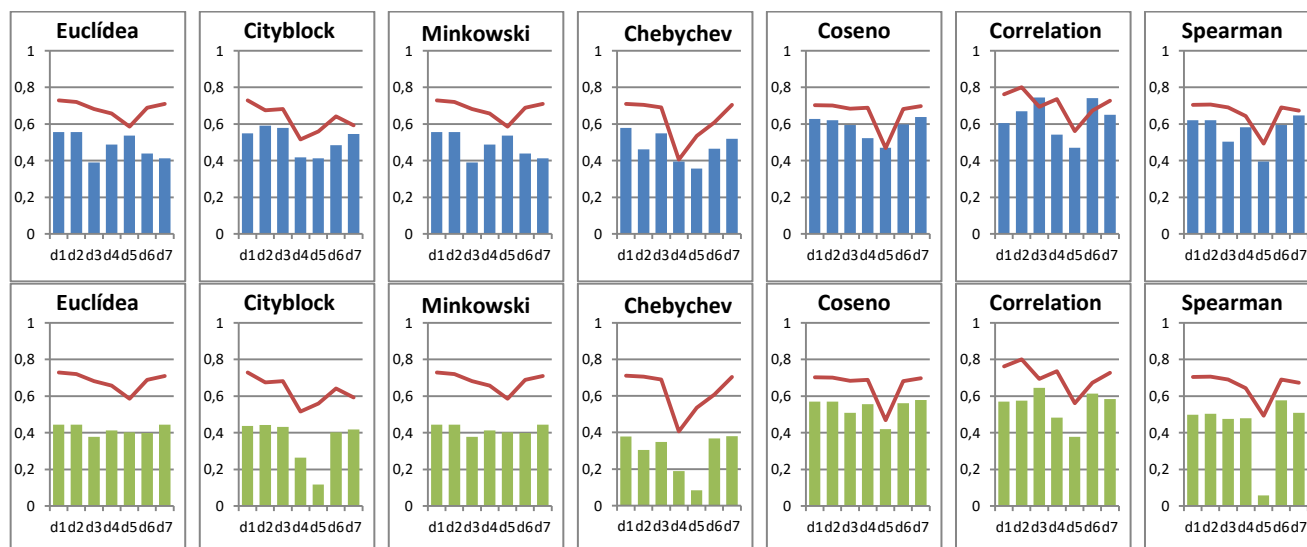
Resultados obtenidos empleando Gist para 8 ventanas y 4 máscaras



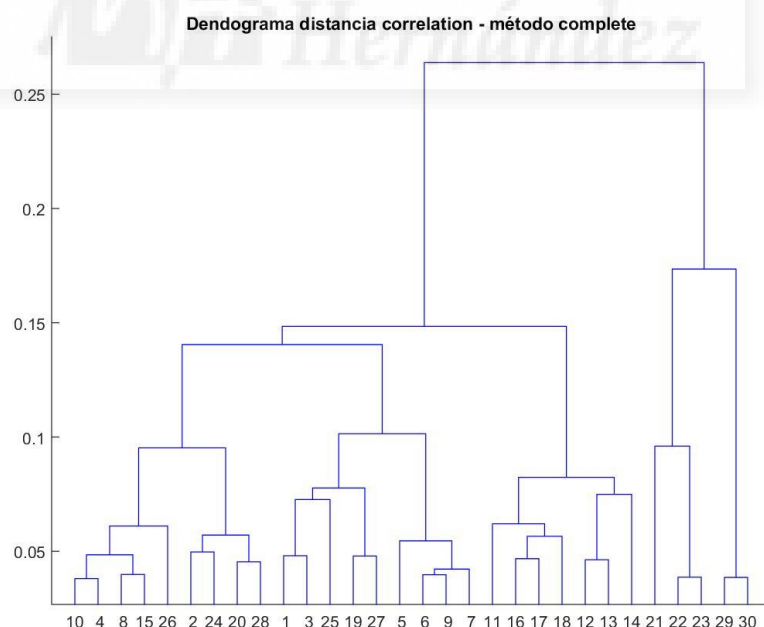
Resultados obtenidos empleando Gist para 8 ventanas y 8 máscaras



Resultados obtenidos empleando Gist para 8 ventanas y 16 máscaras



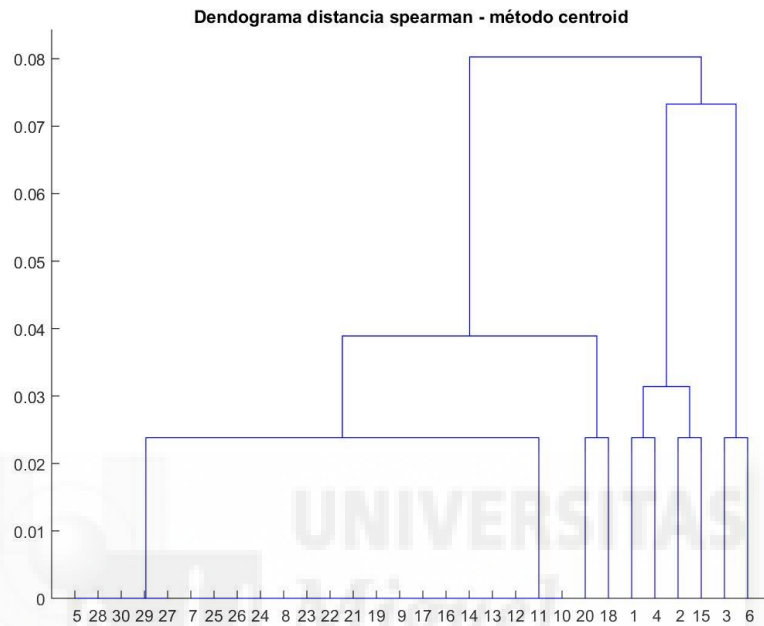
Al emplear Gist como descriptor, se obtienen resultados mejores que con los métodos analizados anteriormente. Apreciamos un acierto medio en torno a 45%-50%. A pesar de ser algo más bajo que el acierto medio en HOG, se obtienen valores concretos más elevados. En este caso, el aumento del número de ventanas y del número de máscaras sí que influye positivamente en los resultados. El mayor acierto se alcanza para 8 ventanas y 8 máscaras empleando la distancia *Correlation* y el método *Complete* con un acierto del 81.95%. El dendrograma obtenido para este caso se muestra a continuación



En cuanto los valores obtenidos de la función *NMI*, la mayoría de los resultados se encuentra alrededor del 30%-45%. El valor más elevado para *NMI* se consigue para el mismo caso con el que obtenemos el mayor porcentaje de acierto, 8 ventanas y 8 máscaras

empleando la distancia *Correlation* y el método *Complete* con un valor de 69.71%.

La media del coeficiente de correlación cofenético se encuentra en torno a un 0.60-0.65. El coeficiente de correlación cofenético correspondiente al caso de 8 ventanas y 8 máscaras es de 0.772. El mejor resultado, 0.870, se obtiene con 1 ventana y 4 máscaras empleando la distancia *Spearman* y el método *Centroid*. El dendrograma obtenido para este caso se muestra a continuación.



En general, igual que con HOG, los mejores resultados de la función *success* y de la función *NMI* se obtienen con los tipos de distancia *Coseno* y *Correlation*. Los peores resultados para la función *NMI* se obtienen con el tipo de distancia *Chebychev* y con el método *Single*.

Como conclusión, tanto HOG como Gist ofrecen mejores resultados que la Firma de Fourier tanto en porcentaje de acierto, como en precisión y correlación a pesar de que en ninguno de los casos, se ha obtenido un acierto del 100%.

Se debe tener en cuenta que las imágenes fueron tomadas mientras el robot seguía una trayectoria. Debido a esto, no se tiene rastreada la planta entera y no se dispone de un mapa denso, por lo que quedan zonas de la planta sin capturar. Esto supone que sea más dificultoso agrupar las imágenes ya que hay muchas imágenes de algunas zonas y pocas de otras dado que el robot también varía su velocidad de desplazamiento. De cara a un futuro problema de localización, también habría que tener en cuenta esta limitación porque el robot podría alcanzar una posición de la planta que no haya sido rastreada previamente.

Además, en las imágenes empleadas nos encontramos con el problema de aliasing visual. Hay un gran número de imágenes de puertas que corresponden a lugares diferentes y que son confundidas por el algoritmo y agrupadas en un mismo cluster. Principalmente ocurre con las imágenes referentes al pasillo y al baño, las cuales, la mayoría de las veces, son agrupadas en un mismo cluster.

A continuación se muestra la representación del mapa de la planta obtenido mediante clustering jerárquico indicando al algoritmo que se desean obtener 5 clusters. Cabe destacar, que las posiciones en las que han sido tomadas las imágenes no se han empleado en ninguno de los procesos, únicamente se ha hecho uso de ellas para plotear este mapa asignando en cada posición un color en función de la etiqueta otorgada por la función *cluster*. La siguiente Figura 5.8 muestra los resultados obtenidos empleando Gist como descriptor con 8 ventanas y 8 máscaras, con la distancia *Correlation* y el método *Complete*, configuración con la cual se han alcanzado los valores más altos de acierto y precisión.

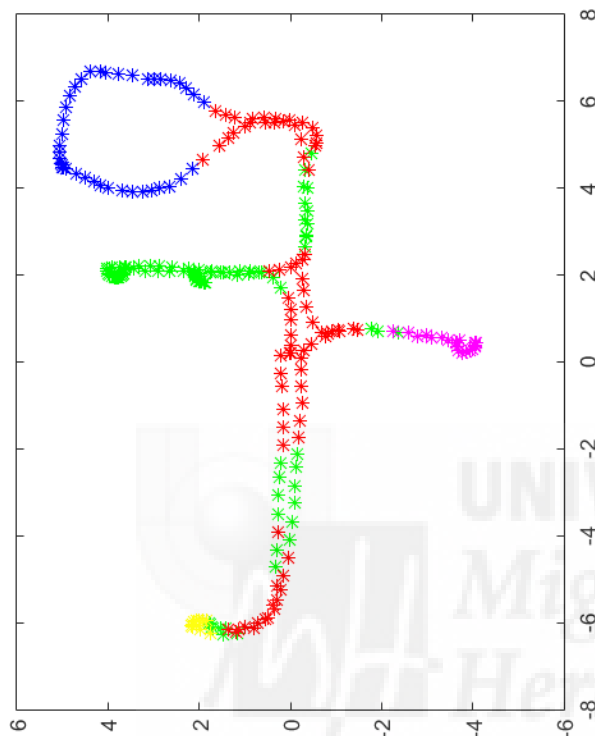


Figura 5.8. Mapa en planta obtenido mediante clustering jerárquico con acierto del 81.95%

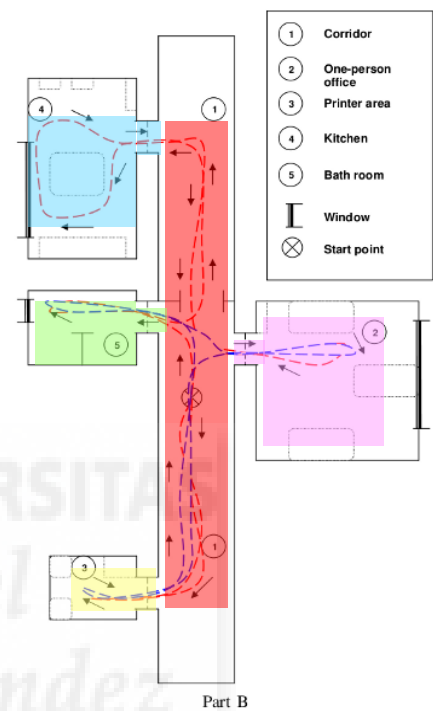


Figura 3.9. Límites de cada estancia

Las imágenes de la cocina, del despacho y de la sala de impresoras han sido agrupadas correctamente, sin embargo, como se comentaba, para la mayoría de casos, se agrupan las imágenes del baño y del pasillo en un mismo cluster. Para conseguir un resultado con un mayor porcentaje de acierto, se va a tratar de distinguir, en la siguiente fase, las imágenes del pasillo y del baño, representadas en verde, con el algoritmo de clustering espectral. Del mismo modo se pretenderá conseguir un mapa de nivel intermedio de las estancias que han sido agrupadas en un mismo cluster en esta primera fase.

Se ha estudiado también la posibilidad de aumentar el número de clusters en esta primera fase de clustering jerárquico tratando de destinar 2 de ellos al pasillo con el objetivo de disponer de un cluster más que pudiera ser destinado para el baño. Tras analizar los resultados, se ha comprobado que no han sido favorables y ese cluster extra se destina, en la mayoría de casos, para dividir la cocina en 2 conjuntos.

5.6.2. Segunda fase: Clustering espectral

En esta segunda fase se pretende conseguir un mapa de nivel medio en el que se realice una nueva agrupación a un nivel más detallado para cada uno de los clusters generados con el clustering jerárquico.

Además, dado que el mejor de los resultados obtenidos con el clustering jerárquico es de un 81.95% de acierto, se va a emplear el clustering espectral para tratar de agrupar correctamente las imágenes correspondientes al pasillo y al baño.

En primer lugar, generamos la matriz de similitud mediante la función *genn_nn_distance*. En este caso, no se pasará a la función la totalidad de las imágenes sino que se pasarán aquellas que han sido etiquetadas con el mismo número en la fase anterior. Por tanto habrá que seleccionar los descriptores correspondientes a las imágenes que formen un mismo cluster, es decir, nos quedaremos con las columnas de la matriz de descriptores correspondientes a las imágenes que hayan sido etiquetadas con un mismo número.

El siguiente paso es agrupar las imágenes mediante la función *sc* empleando la matriz dispersa obtenida en el paso anterior. Esta función, devuelve un vector T de etiquetas que corresponden al número de cluster asignado a cada imagen, en función del número de clusters que se indican que se desean obtener.

El primer cluster con el que vamos a tratar es el que aparece en la Figura 5.8 representado en color verde. En este cluster se han agrupado imágenes tanto del pasillo como del baño. En una primera prueba indicamos que deseamos obtener 2 clusters y el resultado obtenido es el que se muestra en la Figura 5.10.

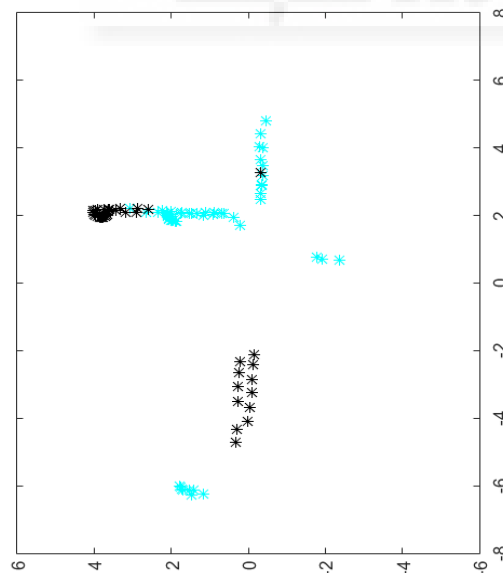


Figura 4.10. Resultado obtenido para 2 clusters

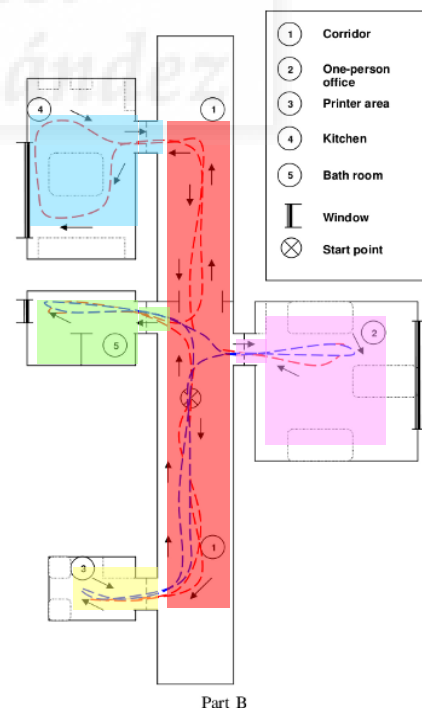


Figura 5.11. Límites de cada estancia

Se observa que el agrupamiento realizado no es correcto ya que vuelve a juntar escenas de dos estancias diferentes en un mismo cluster.

Dado que se sigue persiguiendo el objetivo de separar en clusters diferentes estas imágenes, se realizan varias pruebas incrementando en cada una de ellas el número de clusters.

Para 3 clusters, el resultado obtenido es similar al de 2 clusters. Para 4 clusters se obtiene un resultado bastante mejor, que se muestra en la Figura 5.12.

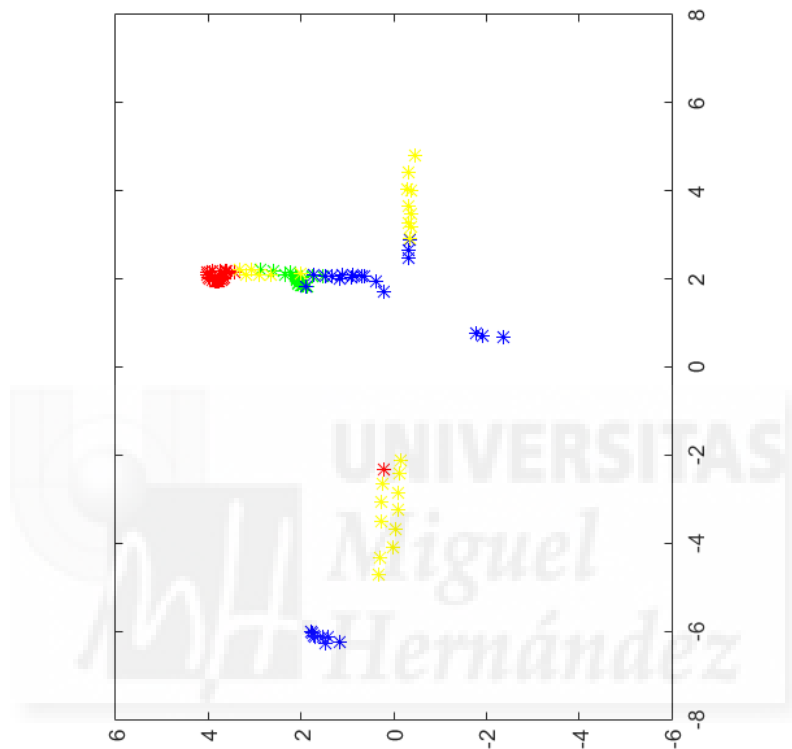


Figura 5.12. Resultado obtenido para 4 clusters

Se consigue separar casi por completo las escenas obteniendo 2 clusters para el baño (rojo y verde) y 2 para el pasillo (azul y amarillo). Aun así se comete un pequeño error en el cluster representado en color amarillo en la Figura 5.12.

Si juntamos estos resultados con los obtenidos en la primera fase de clustering jerárquico, tendríamos 4 clusters clasificados correctamente correspondientes a cocina, despacho, zona de impresoras y parte del pasillo y, por otro lado 4 clusters clasificados prácticamente correctamente también, 2 de ellos correspondientes al baño y otros 2 correspondientes al pasillo. Si representamos los 2 clusters pertenecientes al baño de un mismo color y los 2 correspondientes al pasillo en otro color obtenemos el mapa que se muestra en la Figura 5.13. Para este caso final, el porcentaje de acierto es de un 88.72%.

Como se puede apreciar en la Figura 5.13, la entrada de cada una de las estancias se identifica como pasillo. Una persona también podría considerar visualmente que esas imágenes corresponden al pasillo, pero para ser rigurosos se han establecido los límites de cada estancia en el lugar donde se encuentra la puerta. Por tanto esta agrupación ya se podría

considerar acertada. El plano ideal con el que se realizan las comparaciones para hallar el porcentaje de acierto se muestra en la Figura 5.14.

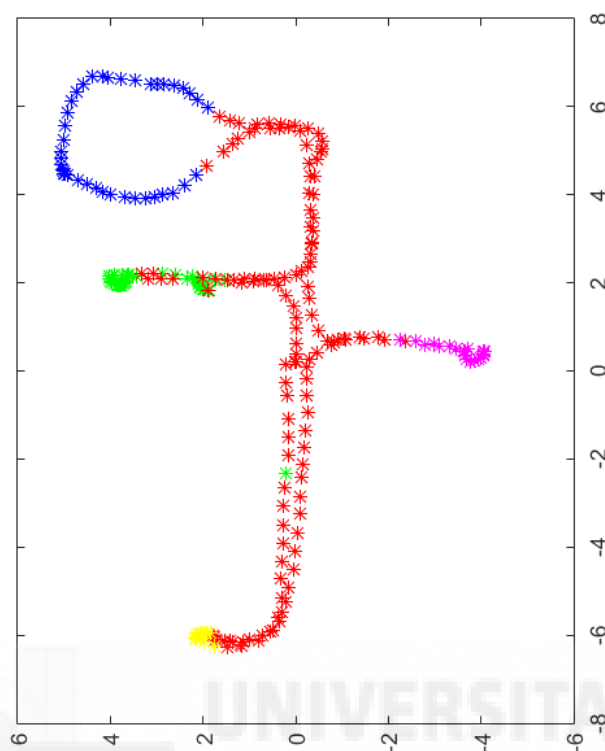


Figura 5.13. Mapa en planta con acierto del 88.72%

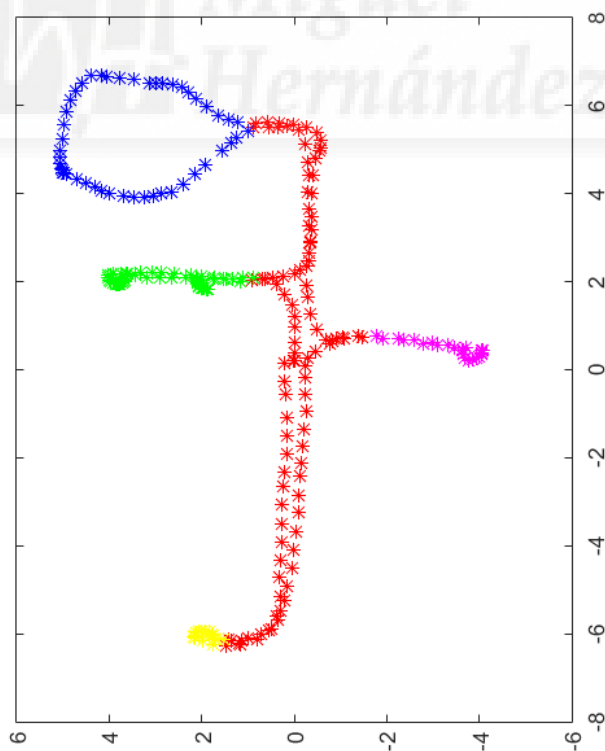


Figura 5.14. Mapa en planta ideal de acierto 100%

Ahora, el objetivo es obtener un mapa de nivel intermedio de alguna de las estancias. Como se puede observar en el mapa de la planta, la única habitación que se recorre captando más detalles es la cocina, ya que para las otras estancias el robot únicamente entra, gira 180° y sale. Por esta razón, se va a realizar el mapa de nivel intermedio para dicha estancia.

En la cocina se pueden distinguir tres zonas, una que se encuentra al lado derecho de la mesa central donde se encuentran los utensilios de cocina y los fogones la cual tiene la pared recubierta de azulejos, otra donde se encuentra un ventanal y otra donde la pared es lisa y hay una mesa auxiliar.

Al realizar la agrupación mediante el algoritmo de clustering espectral para esta estancia y 3 clusters, se obtiene el resultado mostrado en la Figura 5.15, el cual podría decirse que se ajusta con bastante precisión a la realidad.

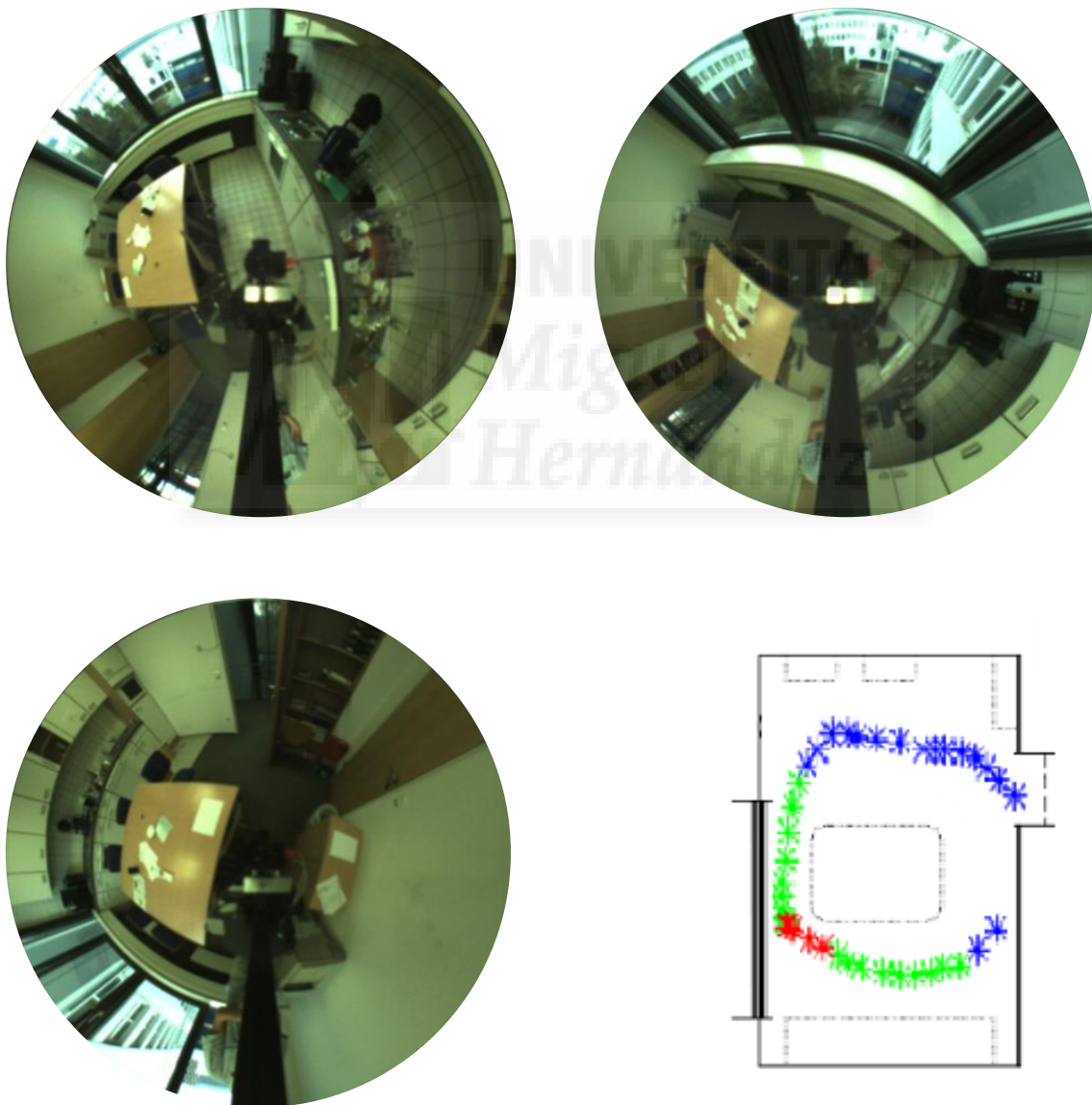


Figura 5.15. Imágenes de las 3 zonas de la cocina. Resultado del clustering espectral.

5.6.3. Única fase: Clustering espectral

Se van a realizar dos experimentos empleando el clustering espectral. En primer lugar, se pretende conseguir un mapa de alto nivel en el que los clusters obtenidos se correspondan con las diferentes estancias de la planta empleando únicamente el clustering espectral. En segundo lugar se quiere obtener un mapa de nivel intermedio directamente empleando también este tipo de clustering.

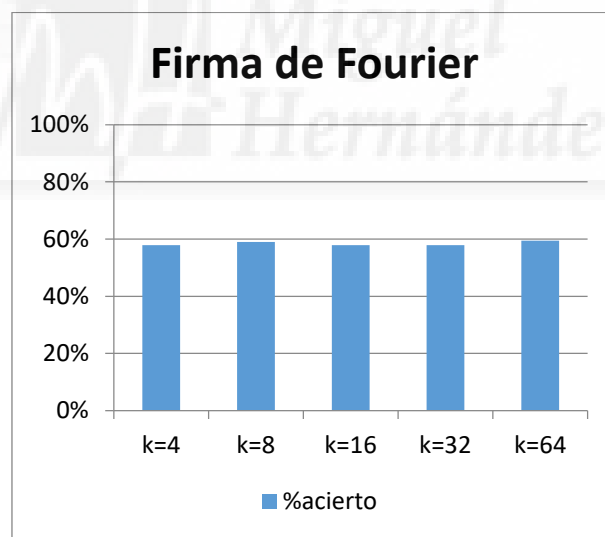
En el clustering jerárquico, el procedimiento a seguir es el explicado en el apartado anterior. En primer lugar se genera la matriz de similitud mediante la función `genn_nn_distance`. En este caso, sí se pasará a la función la totalidad de las imágenes.

El siguiente paso es agrupar las imágenes mediante la función `sc` empleando la matriz dispersa obtenida en el paso anterior. Esta función, devuelve un vector `T` de etiquetas que corresponden al número de cluster asignado a cada imagen, en función del número de clusters que se indican que se desean obtener.

En un primer estudio, se va a indicar a la función `sc` que se desean obtener 5 clusters, los correspondientes al número de estancias de la planta.

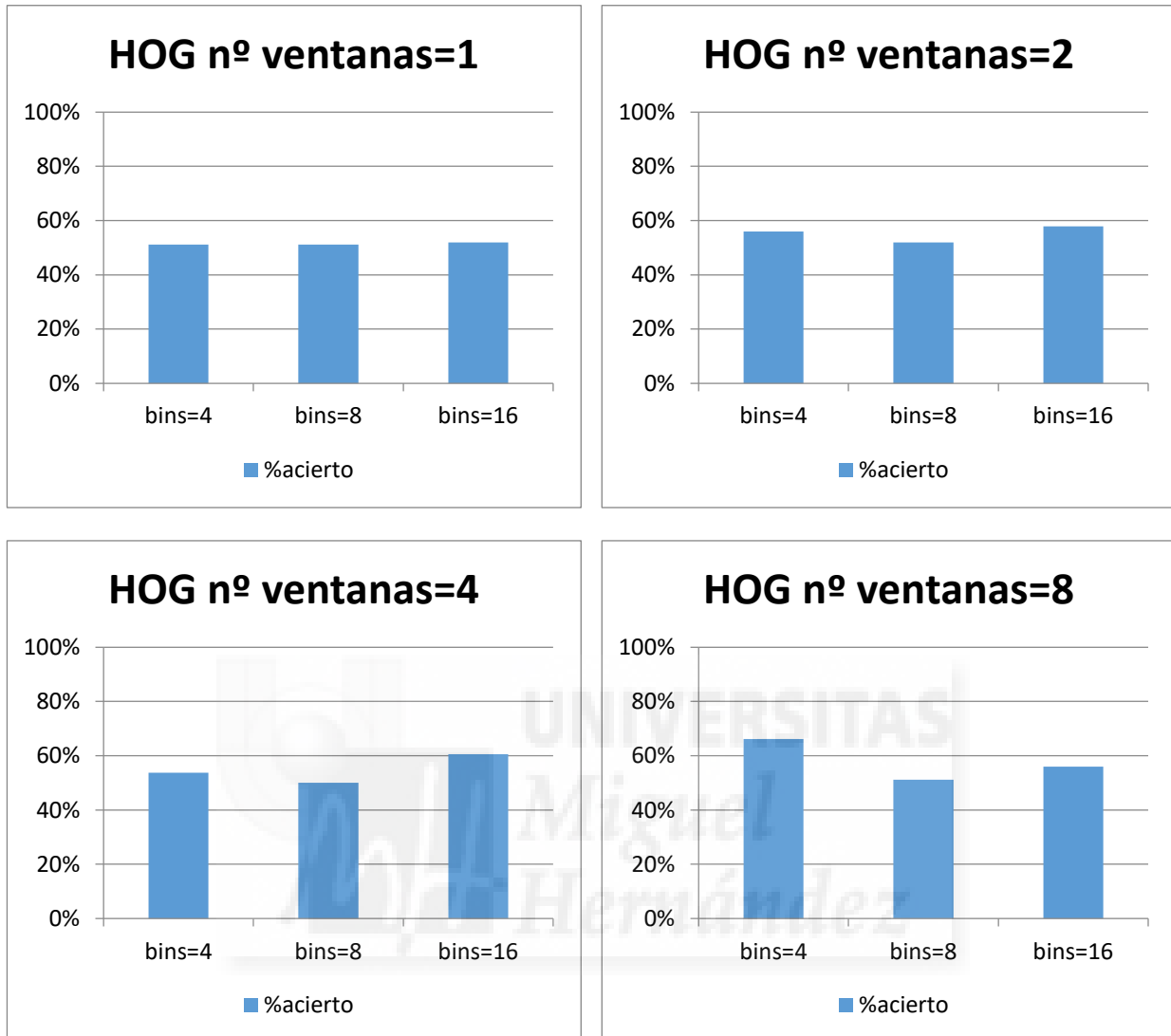
A continuación, se muestran las gráficas en las que se indica el porcentaje de acierto calculado mediante la función `success`.

Resultados obtenidos empleando como descriptor la Firma de Fourier:



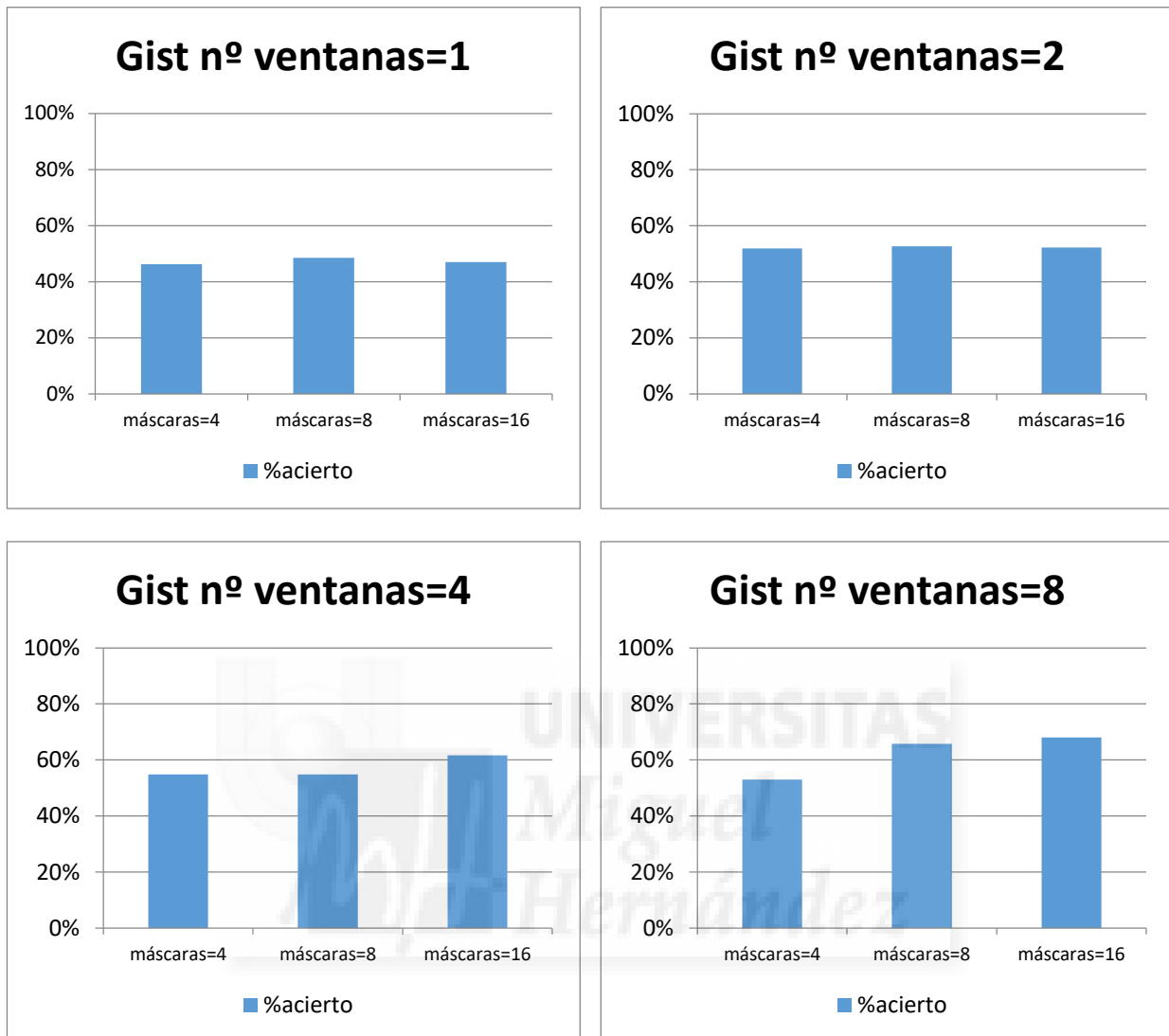
Al emplear la Firma de Fourier como descriptor, los resultados del clustering espectral están próximos al 60%. A pesar de aumentar el valor de `k`, los resultados siguen siendo prácticamente iguales. El mejor resultado es de 59.4% y se alcanza para `k=64`. Este porcentaje de acierto se encuentra todavía muy alejado de un resultado aceptable.

Resultados obtenidos empleando como descriptor HOG:



En este caso, al emplear HOG, el porcentaje de acierto se queda en torno al 50% para 1 ventana, independientemente del número de bins. Para 2 ventanas el acierto es algo mayor, llegando a alcanzarse un 58% con 16 bins. Para 4 ventanas, el acierto se queda por debajo del 55% para 4 y 8 bins pero se alcanza un acierto del 61% para 16 bins. Por último, para 8 ventanas, se consigue un acierto del 66.2% con 4 bins, siendo este el mejor de los resultados obtenidos con este descriptor.

Resultados obtenidos empleando como descriptor Gist:



Empleando Gist como descriptor, vemos que el acierto no llega a alcanzar el 50% para 1 ventana y para 2 ventanas ronda el 52%, independientemente del número de máscaras. Con 4 ventanas se obtiene un acierto del 55% con 4 y 8 máscaras y un acierto del 61.7% para 16 máscaras. Con 8 ventanas se consigue un acierto del 53% para 4 máscaras, 65.8% para 8 máscaras y 68% para 16 máscaras.

Por tanto, se observa que mediante el clustering espectral en 1 fase, no se consigue agrupar correctamente las imágenes en estancias diferenciadas.

Como alternativa de mejora, se propone realizar un agrupamiento mediante clustering espectral pero en varias fases, es decir, comenzar creando clusters más grandes que contengan elementos de varias estancias pero cuyo porcentaje de acierto sea cercano al 100% e ir dividiendo sucesivamente cada cluster en los correspondientes a las estancias buscando también siempre un acierto lo más elevado posible.

En primer lugar se realiza una división en 2 clusters. La configuración de descriptores con la que mayor acierto se consigue al dividir en 2 clusters es con HOG para 8 ventanas y 16 bins. Este será el punto de partida del clustering. El resultado obtenido se muestra a continuación:

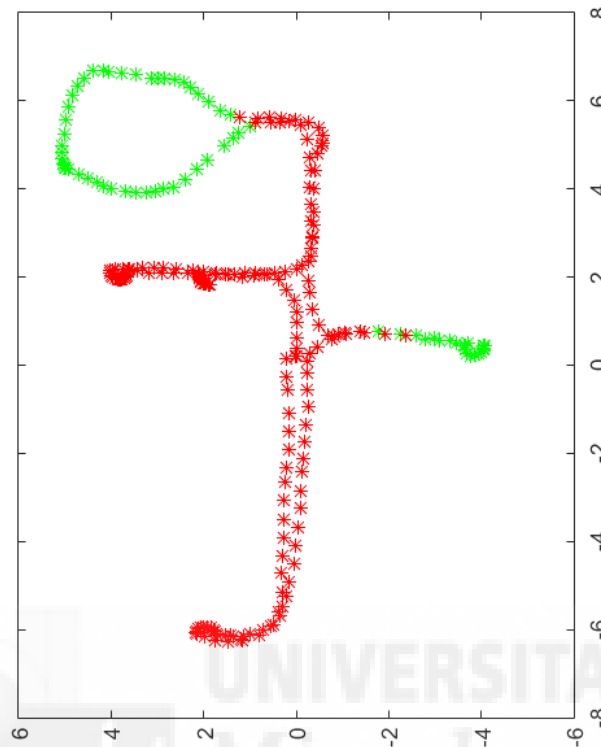


Figura 5.16. Resultados obtenidos del clustering espectral con 2 clusters

Se puede apreciar que se clasifican correctamente las escenas en 2 grupos. Uno que contiene las imágenes correspondientes a la cocina y el despacho y otro que contiene las imágenes del pasillo, del baño y del cuarto de las impresoras.

En la siguiente fase, se van a dividir los clusters creados (verde y rojo en la Figura 5.16). El número de clusters que mejor agrupa las escenas representadas en verde es 3. Se obtienen 2 clusters correspondientes a la cocina y 1 correspondiente al despacho. El resultado se muestra en la Figura 5.17. Sólo 1 escena estaría mal clasificada.

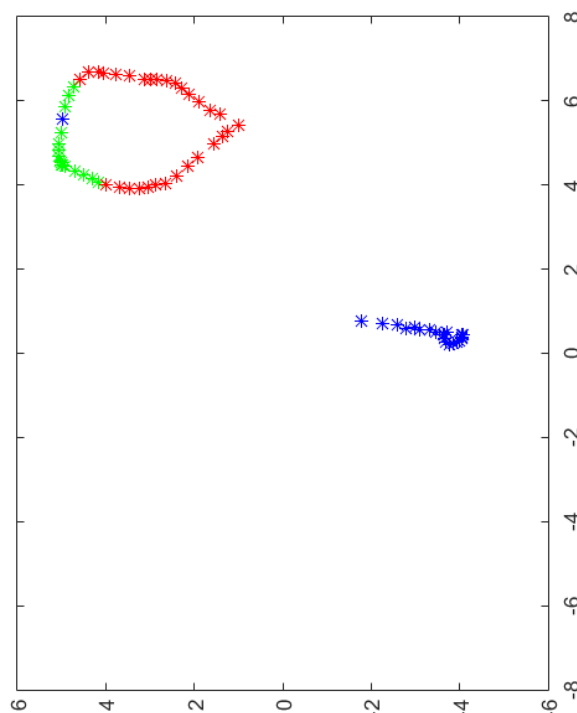


Figura 5.17. Resultado de la agrupación en 3 clusters para la segunda fase

El número de clusters que mejor agrupa las escenas representadas en rojo es también 3. Se obtienen 2 clusters que contienen imágenes del pasillo y del baño y 1 correspondiente al cuarto de las impresoras. El resultado se muestra en la Figura 5.18. El porcentaje de acierto en este caso también es muy elevado.

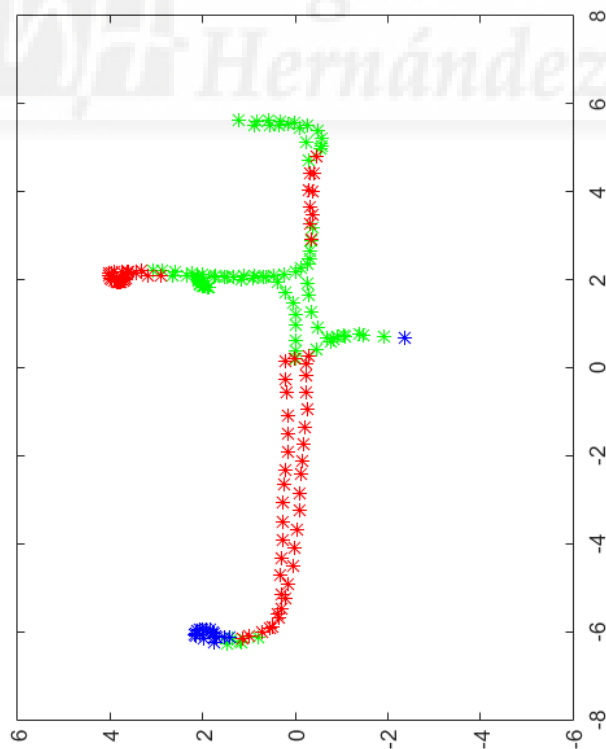


Figura 5.18. Resultado de la agrupación en 3 clusters para la segunda fase

Nos encontramos ahora en una situación muy similar a la que teníamos tras realizar el agrupamiento mediante el clustering jerárquico, escenas del baño y del pasillo agrupadas en un mismo cluster. En este caso, el número de clusters que mejor agrupa las escenas representadas en rojo en la Figura 5.18 es 5. Se obtienen 2 clusters correspondientes al baño y 3 correspondientes al pasillo. El resultado es el que se muestra en la Figura 5.19.

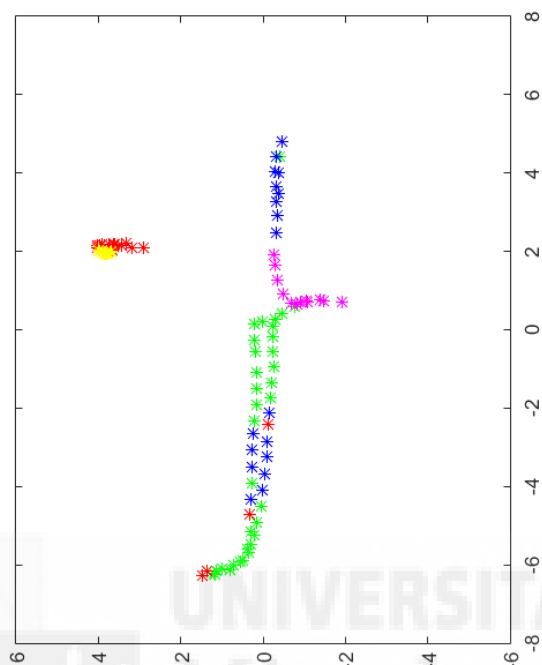


Figura 5.19. Resultado de la agrupación en 5 clusters para la tercera fase

Para las escenas correspondientes al baño y al pasillo representadas en verde en la Figura 5.18 el mejor agrupamiento se consigue con 2 clusters. El resultado se muestra en la Figura 5.20.

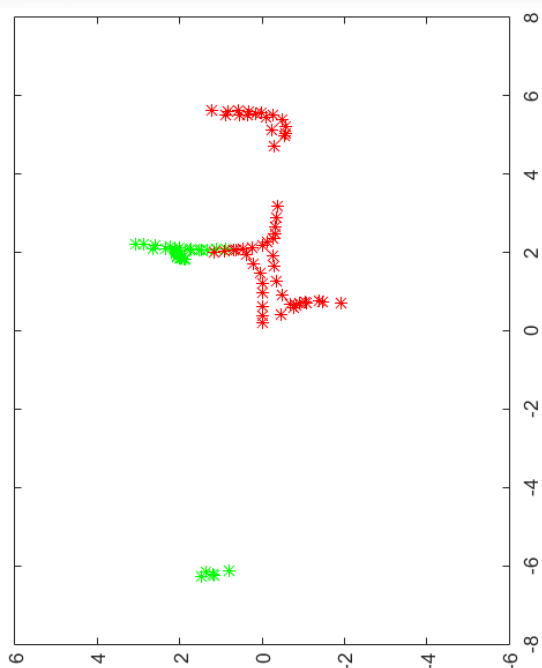


Figura 5.20. Resultado de la agrupación en 2 clusters para la tercera fase

Si representamos los 3 clusters pertenecientes al baño de un mismo color, los 4 correspondientes al pasillo en otro color y los 2 correspondientes a la cocina en otro color, obtenemos el mapa que se muestra en la Figura 5.21. Para este caso final, el porcentaje de acierto es de un 94.74%. Este resultado es muy acertado. Al igual que para el clustering jerárquico, el plano ideal con el que se realizan las comparaciones para hallar el porcentaje de acierto se muestra en la Figura 5.14.

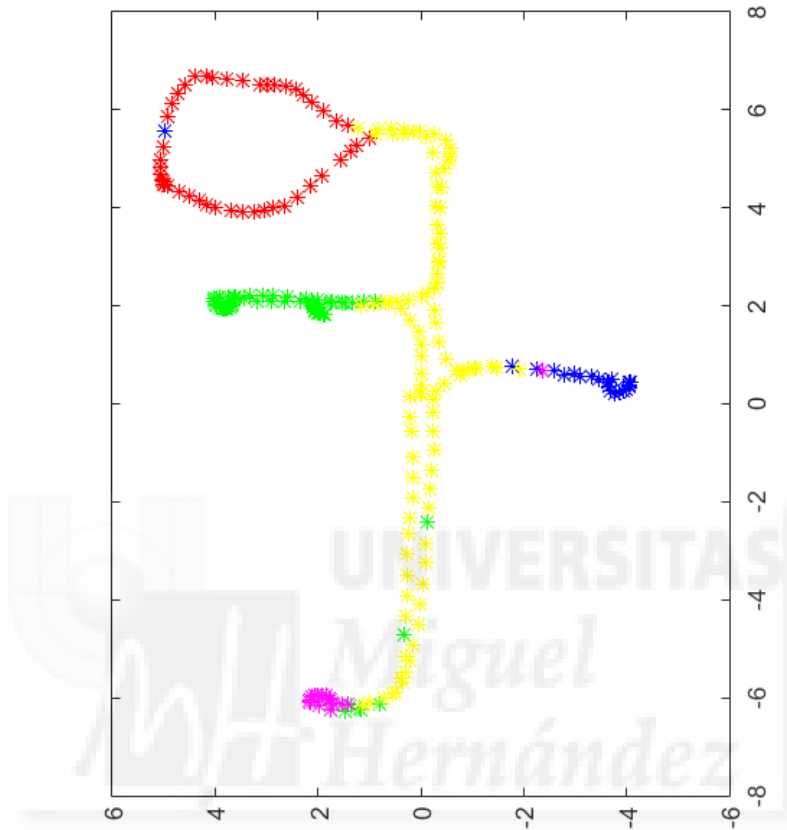


Figura 5.21. Resultado final del clustering espectral con acierto del 94.74%

Una vez analizados los resultados de generar el mapa de alto nivel empleando clustering espectral, vamos a estudiar la posibilidad de generar directamente un mapa topológico de nivel intermedio aumentando el número de clusters.

Este estudio se va a realizar empleando los descriptores que mejores resultados han proporcionado al realizar la agrupación en 5 clusters. Éstos son:

Firma de Fourier: $k=8$ y $k=64$.

HOG: 4 ventanas y 16 bins, 8 ventanas y 4 bins.

Gist: 8 ventanas y 8 máscaras, 8 ventanas y 16 máscaras.

El número de clusters que se va a indicar a la función sc serán: 10, 15, 20, 25 y 30.

A continuación se muestran y se comentan los resultados obtenidos.

Para la agrupación en 10 clusters, los resultados que se obtienen utilizando la Firma de Fourier como descriptor son erróneos. Hay un gran número de imágenes mal clasificadas. Al emplear como descriptor HOG, los resultados mejoran bastante en comparación a descriptor anterior. Para el caso de 4 ventanas y 16 bins se obtiene la mejor clasificación de todas las analizadas al generar 10 clusters. El resultado de esta agrupación se muestra en la Figura 5.22. Utilizando Gist como descriptor, los resultados obtenidos son mejores que para la Firma de Fourier y peores que los obtenidos con HOG.

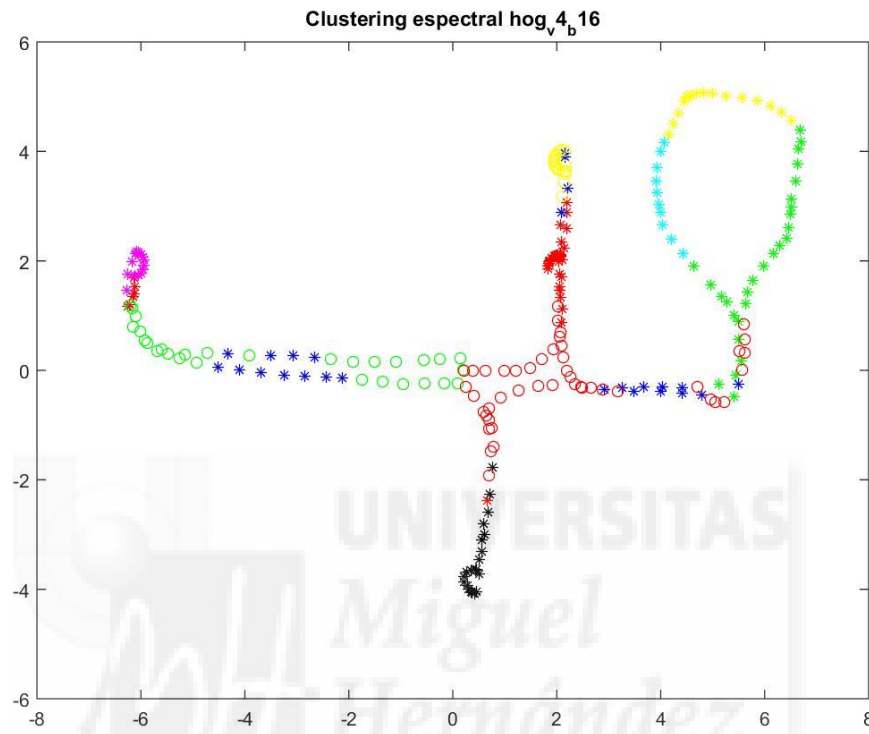


Figura 5.22. Resultado obtenido mediante agrupación en 10 clusters. HOG 4 ventanas y 16 bins

Para la agrupación en 15 clusters, de nuevo obtenemos las peores agrupaciones empleando la Firma de Fourier. Los resultados son algo mejores que los obtenidos en la agrupación en 10 clusters pero aun así tenemos un gran número de escenas mal agrupadas. El descriptor Gist proporciona unos resultados que también presentan algo de error pero son mejores que los de la Firma de Fourier. Al igual que en el caso anterior, los mejores resultados se obtienen empleando HOG como descriptor. En la Figura 5.23 se muestran los resultados obtenidos para el caso en el que se tiene el mayor número de escenas correctamente clasificadas. De nuevo vuelve a ser para 4 ventanas y 16 bins, en cambio el resultado obtenido para 10 clusters era algo más acertado.

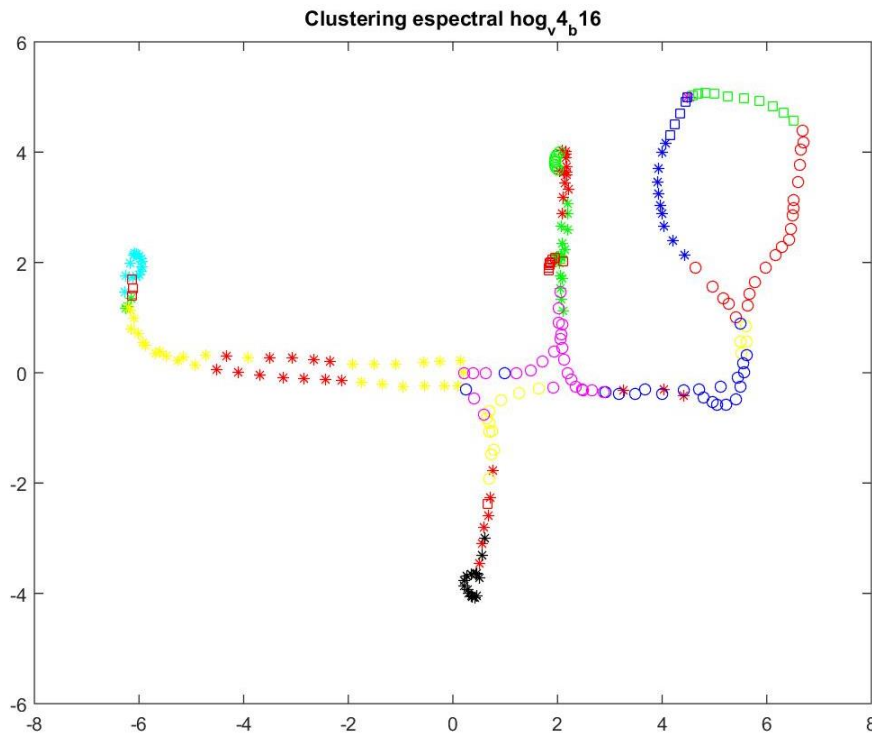


Figura 5.23. Resultado obtenido mediante agrupación en 15 clusters. HOG 4 ventanas y 16 bins

Para el caso de la agrupación en 20 clusters, con la Firma de Fourier se siguen obteniendo escenas mal agrupadas. Para HOG y para Gist los resultados son mejores, siendo algo mejores los obtenidos con Gist. En la Figura 5.24 se muestra el resultado obtenido empleando Gist como descriptor con 8 ventanas y 8 máscaras, configuración con la que mejor agrupación hemos obtenido.

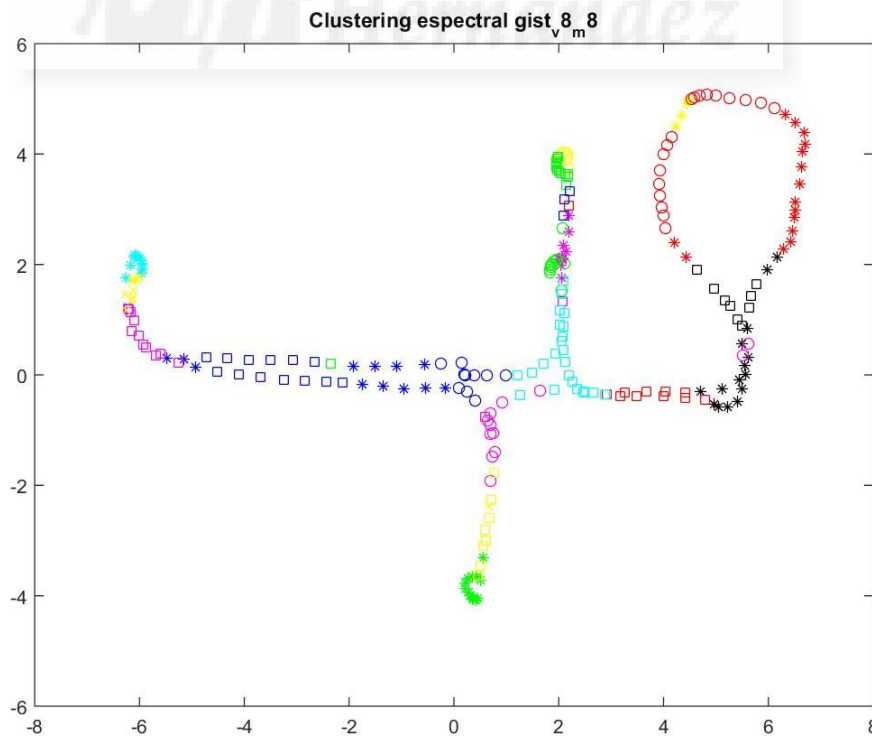


Figura 5.24. Resultado obtenido mediante agrupación en 20 clusters. Gist 8 ventanas y 8 máscaras

Para la agrupación en 25 cluters, la Firma de Fourier sigue sin ofrecer un buen resultado y tanto para Gist como para HOG los resultados presentan un alto porcentaje de acierto en la agrupación. En este caso, la configuración que mejor resultado da es empleando Gist con 8 ventanas y 16 máscaras, aunque con la otra configuración de Gist y las dos de HOG estudiadas los resultados son muy similares. En la Figura 5.25 se muestra la agrupación obtenida.

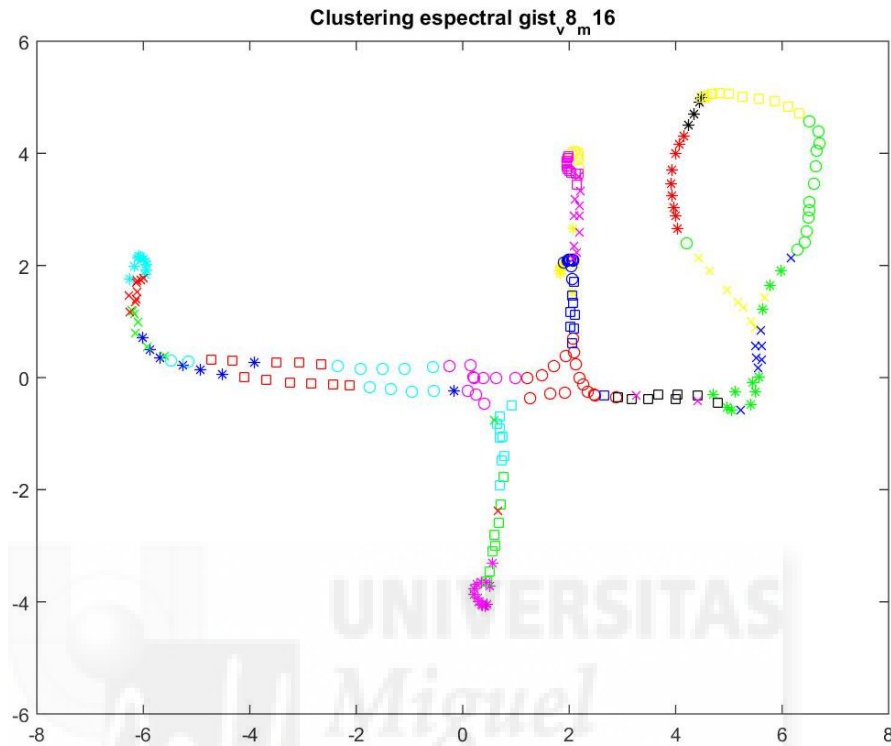


Figura 5.25. Resultado obtenido mediante agrupación en 25 clusters. Gist 8 ventanas y 16 máscaras

Para la agrupación en 30 clusters, tampoco se ha conseguido un buen resultado para la Firma de Fourier. Los resultados obtenidos con HOG son mejores y los obtenidos con Gist los más acertados. Aun así, el hecho de haber aumentado el número de clusters no ha supuesto una mejora frente a otras agrupaciones en un menor número de conjuntos. En la Figura 5.26 se muestra uno de los resultados obtenidos con Gist.

En conclusión, para el caso de esta base de datos, la Firma de Fourier no ofrece buenos resultados con el clustering espectral. Los resultados obtenidos con HOG y con Gist son suficientemente acertados a pesar de que con ninguno de ellos se obtiene una clasificación perfecta. También se ha observado que a pesar de que se aumente el número de clusters no tiene porqué mejorar el resultado. De todas las configuraciones estudiadas, la más acertada podría decirse que es generando 20 clusters con Gist como descriptor con 8 ventanas y 8 máscaras, a pesar de que evaluar visualmente el número de escenas clasificadas correctamente es bastante difícil cuando las imágenes han sido tomadas siguiendo una ruta.

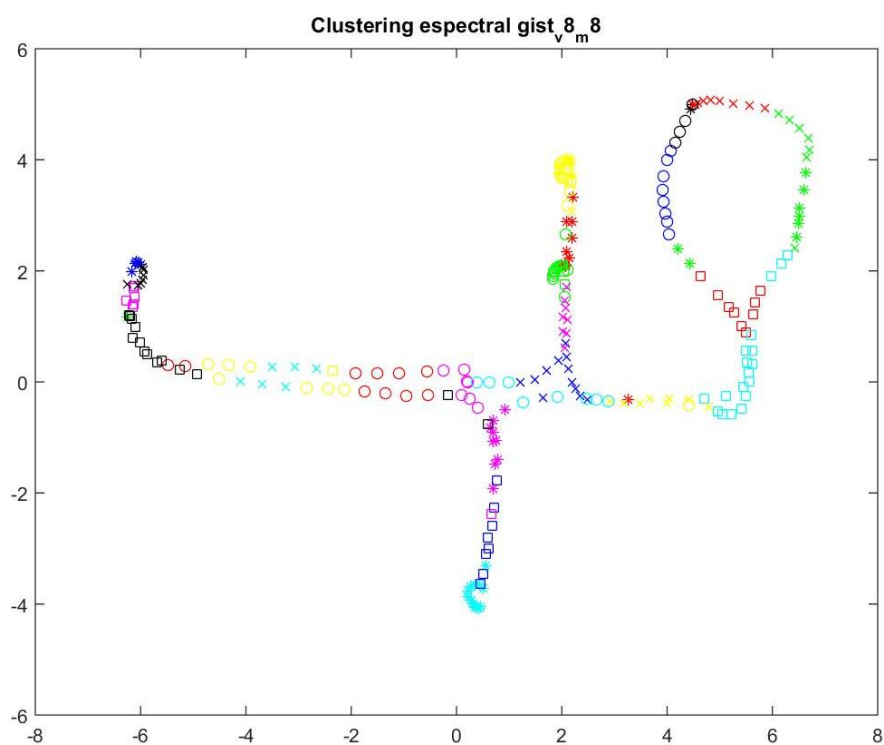


Figura 5.26. Resultado obtenido mediante agrupación en 30 clusters. Gist 8 ventanas y 8 máscaras





Capítulo 6.

Conclusiones

Este trabajo está dedicado al estudio de la creación de mapas visuales topológicos mediante el agrupamiento de escenas. El punto de partida, es un conjunto de imágenes del Centro de Investigación Alemán de Inteligencia Artificial en Saarbrücken contenidas en la base de datos COLD. Estas imágenes han sido captadas por un sensor de visión en formato omnidireccional.

Para extraer la información necesaria de cada una de las imágenes captadas se han empleado diferentes descriptores de apariencia global como la Firma de Fourier, el Histograma de Orientación del Gradiente y Gist con diferentes configuraciones.

De estos descriptores mencionados, se ha estudiado su coste computacional y los resultados obtenidos a la hora de crear el mapa visual topológico al tratar con los diferentes algoritmos de clustering. En cuanto al coste computacional, la Firma de Fourier ofrece unos tiempos muy reducidos en comparación con los otros dos descriptores, siendo Gist el que mayor tiempo ha supuesto.

Se han analizado los resultados del clustering jerárquico y del clustering espectral en sus diferentes configuraciones. Los mejores resultados han sido obtenidos con el descriptor Gist y con los valores más altos de número de ventanas.

Se ha observado que para la base de datos empleada, en la cual las imágenes han sido tomadas siguiendo una ruta, el mejor de los resultados de crear el mapa de alto nivel en una única fase tanto con clustering jerárquico como espectral, no llega a alcanzar el 82% de acierto. Por tanto, se ha optado por realizar el agrupamiento en varias fases. Una de las alternativas ha sido una primera fase de clustering jerárquico seguido de una siguiente fase de clustering espectral y la otra de las alternativas ha sido emplear el clustering espectral en varias fases. De esta manera se ha conseguido alcanzar un acierto próximo al 95%, pudiéndose considerar éste un buen resultado para realizar la tarea de navegación autónoma.

Finalmente, se han analizado los resultados de generar el mapa de nivel intermedio directamente empleando clustering espectral con un mayor número de clusters. Se ha observado que la Firma de Fourier ofrece resultados con bajo grado de acierto mientras que

HOG y Gist han mostrado comportarse adecuadamente teniendo unos resultados de un elevado porcentaje de acierto en la clasificación.

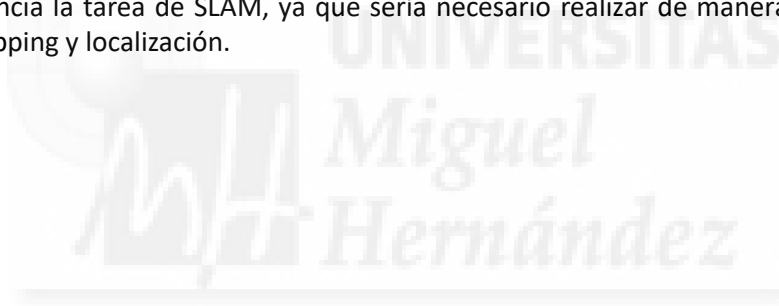
Con los resultados experimentales obtenidos se pueden plantear nuevas líneas de trabajo que permitan avanzar en este campo, ya que se ha comprobado que los descriptores de apariencia global son una alternativa robusta en la extracción de puntos característicos. Entre otras, podemos considerar las siguientes:

- Desarrollar nuevos descriptores de apariencia global que permitan mejorar el porcentaje de acierto en la clasificación de las escenas.

- Mejorar el mapa obtenido ordenando espacialmente los nodos con el método de masa-muelle-amortiguador, un algoritmo basado en un sistema mecánico de fuerzas que permite establecer relaciones topológicas entre las imágenes que forman cada mapa de bajo nivel [24].

- Resolver el problema de localización jerárquica a partir del mapa obtenido. En esta tarea, el robot captura una imagen y la compara en primer lugar con los descriptores representativos de los clusters para detectar en que zona está y posteriormente con las imágenes del cluster para afinar la posición.

- Implementar un sistema de navegación totalmente autónomo mediante algoritmos de control visual que permitan navegar al robot haciendo uso del mapa. En esta línea tomaría gran importancia la tarea de SLAM, ya que sería necesario realizar de manera simultánea las tareas de mapping y localización.



Capítulo 7.

Referencias

- [1] Página web de Matlab: <https://es.mathworks.com/>. Hierarchical clustering
- [2] L. Payá, A.Gil, and O.Reinoso, "A state-of-the-art review on mapping and localization of mobile robots using omnidirectional visión sensors," Journal of Sensors, vol.2017, 2017.
- [3] L. Payá, F. Amorós, L. Fernández, O. Reinoso. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. Sensors. Ed. MDPI. ISSN: 1424-8220. Vol. 14(2), pp. 3033-3064. 2014.
- [4] Berenguer, Y.; Payá, L.; Ballesta, M.; Reinoso, O. "Position Estimation and Local Mapping Using Omnidirectional Images and Global Appearance Descriptors". Sensors 2015, 15, 26368-26395
- [5] E. Menegatti, M. Zocaratto, E. Pagello, and H. Ishiguro. Image-based monte carlo localization with omnidirectional images. Robotics and Autonomous Systems, 48(1):17–30, 2004.
- [6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, pages 886–893, 2005.

- [7] Q. Zhu, S. Avidan, Y. Mei-Chen, and K.T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1491–1498, 2006.
- [8] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(1): 145–175, 2001.
- [9] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155(1):23–36, 2006.
- [10] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007.
- [11] L. Payá. Técnicas de descripción de la apariencia global de escenas: aplicación a la creación de mapas y localización de robots móviles. 2014.
- [12] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, 1988.
- [13] T. Collins, J.J. Collins, and C. Ryan. Occupancy grid mapping: An empirical evaluation. In *Proc. of the 15th Mediterranean Conference on Control & Automation*; Freiburg, Germany, 2007.
- [14] A. Stimec, M. Jogan, and A. Leonardis. Unsupervised learning of a hierarchy of topological maps using omnidirectional images. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(4):639–665, 2008.
- [15] S. Werner, B. Krieg-Brueckner, and T. Herrmann. Modelling navigational knowledge by route graphs. *Spatial Cognition II*, 1849:295–316, 2000.
- [16] A. Angeli, S. Doncieux, J.A. Meyer, and D. Filliat. Visual topological slam and global localization. In *ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 2029–2034, Piscataway, NJ, USA, 2009. IEEE Press.
- [17] L. Fernandez, L. Paya, O. Reinoso, A. Gil, M. Julia, and M. Ballesta. Robust methods for robot localization under changing illumination conditions. Comparison of different

- p>filtering techniques. In Proc. of the Int. Conf. on Agents and Artificial Intelligence; Valencia, Spain, volume 1, pages 223–228, 2010.
- [18] L. Kaufman and P.J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.
- [19] U. Luxburg. A tutorial on spectral clustering. Statistics and computing, 17(4): 395–416, 2007.
- [20] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems, volume 1, pages 849–856, 2001.
- [21] B. Everitt, S. Landau, and M. Leese. Cluster Analysis. Arnold, 5 edition, 2001. ISBN 0-340-76119-9.
- [22] V.Grassi Junior and J.Okamoto Junior, “Development of an omnidirectional visión system”, Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol.28, pp. 58{68,032006.
- [23] A.Pronobis and B.Caputo, “The COLD Database”, Centre for Autonomous Systems, October 2007.
- [24] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. Robotics and Autonomous Systems, 47(4):251–276, 2004.
- [25] L.N. Piotrowsky and F.W. Campbell. A demonstration of the visual importance and flexibility of spatial-frequency amplitude and phase. Perception, 11(1):337– 346, 1982.
- [26] M.J. Morgan, J. Ross, and A. Hayes. The relative importance of local phase and local amplitude in patchwise image reconstruction. Biological Cybernetics, 65 (1):113–119, 1991.
- [27] Página web de la librería de clustering espectral: <https://sourceforge.net/projects/spectralcluster/files/>

